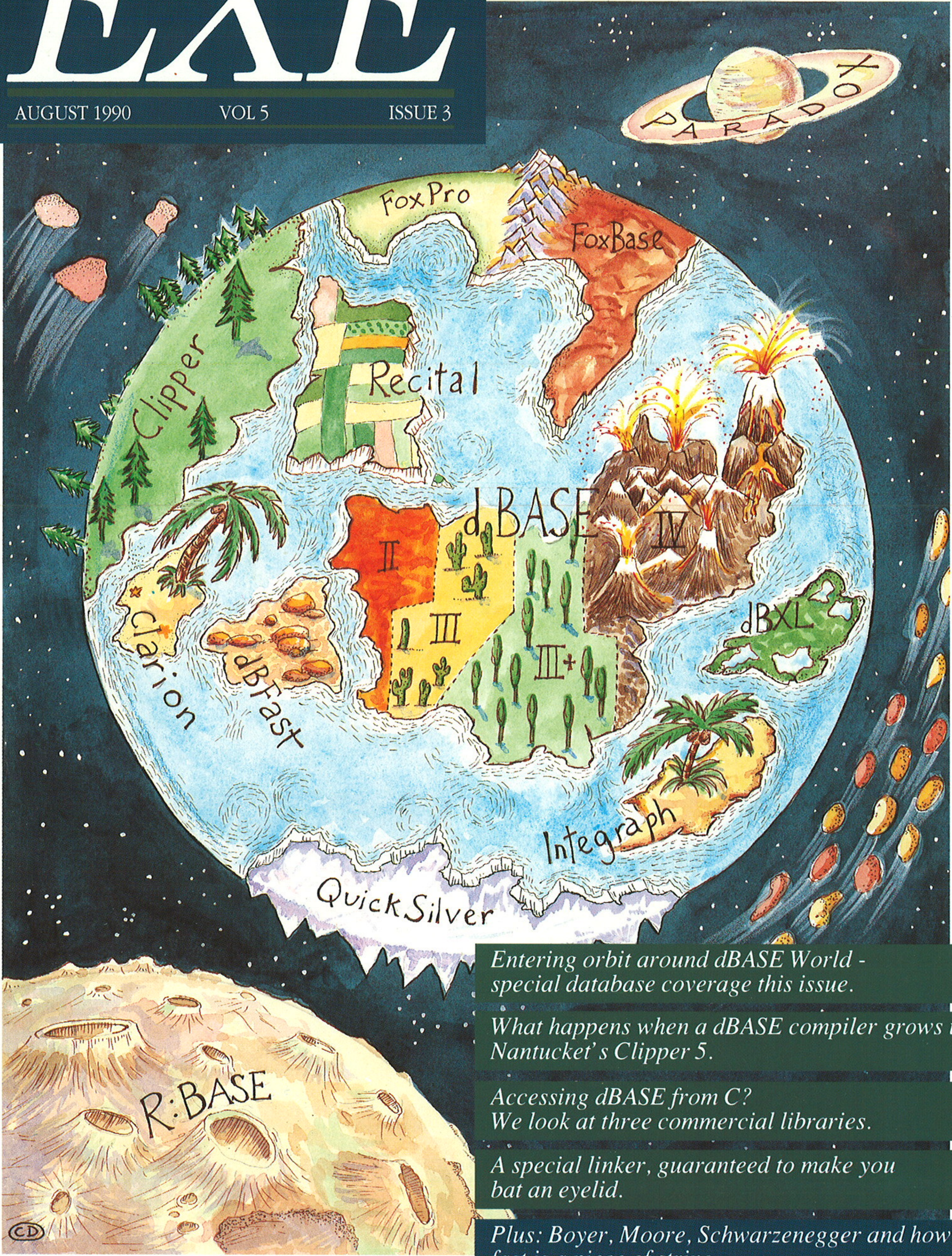
.EXE

AUGUST 1990 VOL 5 ISSUE 3

SCOTTISH OFFICE
1 AUG 1990

£3.20

The Software Developers' Magazine



Entering orbit around dBASE World - special database coverage this issue.

What happens when a dBASE compiler grows up: Nantucket's Clipper 5.

Accessing dBASE from C? We look at three commercial libraries.

A special linker, guaranteed to make you bat an eyelid.

Plus: Boyer, Moore, Schwarzenegger and how fast is a piece of string.

THE TRAINING LEADERS

The best OS/2 training and consultancy...

'Number One in OS/2'

When the major PC manufacturers, corporates and software developers need the most experienced, most knowledgeable, most thorough assistance in working with OS/2, one company comes immediately to mind – QA.

QA's leadership in OS/2 training and consultancy is unique.

We were the first company to be authorised by Microsoft for OS/2 training.

We were the company chosen to be suppliers of OS/2 training and education materials to IBM.

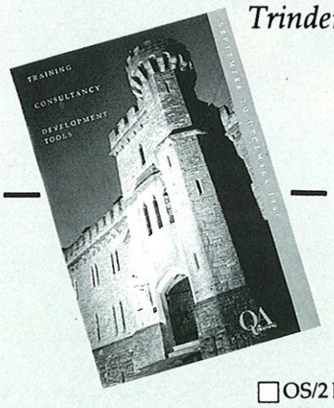
And we were the first company in the world to give Presentation Manager training.

Today, our wide range of courses provide subject-specific training, or progression through a linked series of courses, to give you the knowledge to achieve your particular OS/2 objectives.

And our consultants' expertise and experience can assist you to achieve your design goals without costly mistakes.

So if you are interested in getting the best out of OS/2, why not join those who have discovered that QA is the One to help you.

Send for more details today or call Samantha Trinder on 0285 655888.



OS/2 DEVELOPMENT COURSES

OS/2 PROGRAMMING
4 days £780

OS/2 LAN PROGRAMMING
4 days £860

OS/2 PRESENTATION MANAGER PROGRAMMING
5 days £920

OS/2 APPC PROGRAMMING
4 days £989

OS/2 DEVICE DRIVERS
4 days £1080

OS/2 SQL DATABASE PROGRAMMING
4 days £860

ADVANCED OS/2 PM PROGRAMMING
4 days £980

DIALOG MANAGER PROGRAMMING
3 days £620

...ALSO SUPPORT...

SUPPORTING OS/2 STANDARD EDITION
3 days £580

SUPPORTING OS/2 COMMUNICATIONS MANAGER
3 days £620

SUPPORTING OS/2 LAN MANAGER
4 days £760

INTRODUCTION TO OS/2
2 days £360

SUPPORTING IBM OS/2 AND DOS LANS
4 days £760

...AND SEMINARS...

OS/2 TECHNICAL OVERVIEW
1 day £200

OS/2 SQL DATABASES
1 day £200

OS/2 NETWORKING
1 day £200

OS/2 PRESENTATION MANAGER
1 day £200

...OR CONSULTANCY.

PROJECT ANALYSIS
DESIGN VETTING
DESIGN STANDARDS
PROBLEM SOLVING

Please complete and mail to:
QA Training,
Cecily Hill Castle, Cirencester,
Glos GL7 2EF, ENGLAND

Microsoft® Authorised

IBM Suppliers of OS/2 training and education materials to IBM

Please send me full details on (tick as appropriate):

☐ OS/2 DEVELOPMENT COURSES ☐ OS/2 SUPPORT COURSE ☐ OS/2 SEMINARS
☐ OS/2 CONSULTANCY ☐ OTHER QA COURSES & SEMINARS

NAME
TITLE
COMPANY
ADDRESS
CIRCLE NO. 125
TEL. NO.

QA TRAINING

Window

Help

Vol 5 Issue 3

.EXE

August 1990

Editor: Willie Watts
Editorial Assistant: Wendie Leek
Display Advertising Manager: Sandra Inniss-Palmer
Display Sales Executive: Ed Butcher
Recruitment Sales Executive: Jeremy Cox
Reader Services & Promotions: Helena Adams
Design & Layout: Katerina Adams, Mark English
Office Administrator: Carla Pearce
Publisher: Jon Howell
Front Cover Illustration: Chris Duggan

General
.EXE Magazine is independent and not affiliated to any vendor of hardware, software or services. It is published by Process Communications Ltd, 10 Barley Mow Passage, Chiswick, London W4 4PH.
Tel: (Advertising/Editorial/Production) 081 994 6477
(Subscriptions) 044282 4501
Facsimile: 081 994 1533 Telex: 8811418 SPACES G
ISSN: 0268-6872

Subscriptions
.EXE Magazine is a monthly journal for software developers. It is available only by subscription, at a cost of £35 per annum (11 issues) in the UK (see insert card for academic and overseas rate). 'A subscription implies that this journal will be sent to the subscriber until one of the three expires' - AG Macdonell. The magazine is published around the 26th of the month preceding the cover date. There is no January issue. To subscribe, please call 044282 4501, or write to The Subscriptions Manager, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We can invoice your company or take ACCESS and VISA cards. In case of a query about your subscription, please call 044282 4501.
Back issues are available at a cost of £3.50 each - call our Chiswick office (081 994 6477) for a list of issues still in print. We also offer a range of magazine binders, disk holders etc - please call for details.

Editorial
Editorial enquiries should be addressed to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We welcome letters, opinions, suggestions and articles from our readers. If you are interested in contributing articles, please write to this office for a copy of our Contributors' Guide.
Information contained in .EXE is believed to be correct. If errors are found, we will endeavour to publish a clarification in the next available issue.
From time to time, we offer to copy the PC program code described in an article onto diskette for our readers. In this case, please send a blank, formatted disk with a self-addressed, prepaid mailer to the editorial address given above. We can copy both 5.25" and 3.5" disks.

Advertising
If you are interested in advertising in this magazine, please write to the Display Advertising Manager/Recruitment Sales Manager (as appropriate), .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH, or call 081 994 6477 for details of our advertising rates.

Pronunciation
The name of .EXE Magazine is pronounced to rhyme with 'not sexy magazine'.

Copyright
Material published in .EXE Magazine is copyright © Process Communications Ltd. Articles (or parts of articles) may not be copied, distributed or republished without written permission from the publishers.

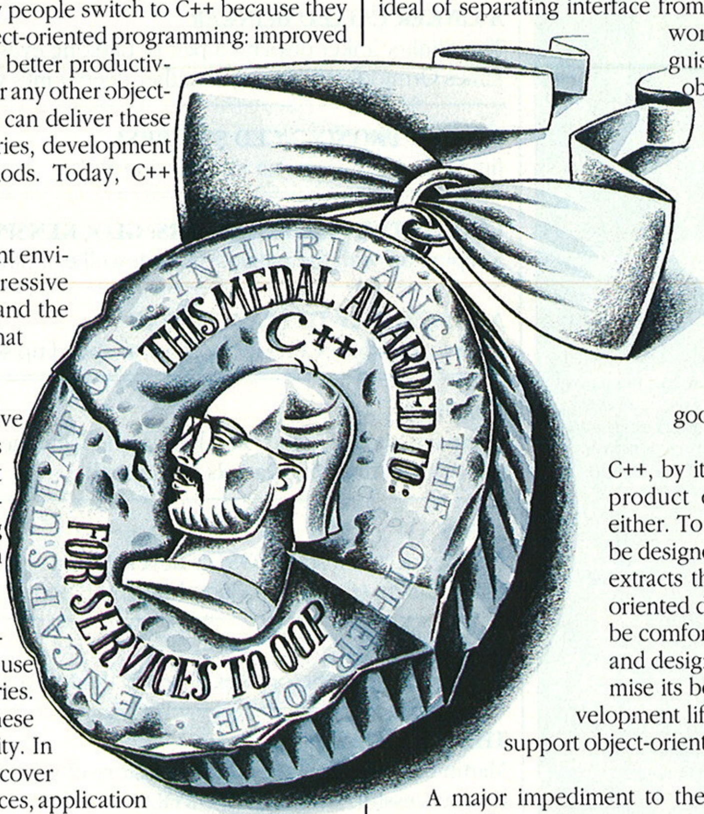
SPECIAL ISSUE - dBASE WORLD

OOP is more than using C++

As the OOP bandwagon moves onwards, developers are increasingly turning to C++ to take advantage of this paridigm. John Daniels doubts the wisdom of this approach.

Thinking of changing to C++? Consider this: during the past year thousands of developers have made the switch and many, perhaps most, are disappointed with the results.

There is nothing wrong with C++ provided you accept it for what it is. A better C. The problem is, many people switch to C++ because they want the benefits claimed for object-oriented programming: improved quality, easier enhancement and better productivity. Unfortunately, neither C++ nor any other object-oriented programming language can deliver these without support from class libraries, development environments and design methods. Today, C++ does not have that support.



The productivity of a development environment depends on the expressive power of a language statement and the time to invent, code and test that statement.

C++ does have more expressive power than C. But, as is always the case, the ability to exploit that power depends upon the richness of the libraries of existing code which we can invoke from primitive statements.

The best way to boost productivity would be to make extensive use of publicly-available class libraries. Today, there aren't many of these and they are often of poor quality. In the main, the libraries available cover three areas: graphical user interfaces, application frameworks and computational abstractions, such as sets, lists, etc. A well-written set of integrated libraries covering all of these areas would go a long way towards improving productivity. At present, the closest we have is a C++ version of Apple's MacApp, the application framework that made ObjectPascal so popular with Macintosh developers. In the UNIX world there is ET++, a public domain library. ET++ is huge, powerful and, ultimately, flawed. For PCs, no integrated application framework is available.

If publicly-available libraries don't have much to offer, we can always look for improved productivity from reuse of our own code. But problems lurk here too. Reuse of code from one project to the next certainly seems feasible, provided time is spent getting it into the correct form. Reuse of code within one project is much harder to achieve, unless the code is being developed and reused by a single person.

The shared reuse of code still being developed is made hard by the primitive state of C++ tools and by the nature of C++ itself. It is a problem of dependencies. C++ class definitions might have public and private sections, but that doesn't mean changes to the private part do not affect users of the class. Far from it. All clients must know, at compile time, the memory size of instances of the class. This means that all client code must be re-compiled whenever data members are added or removed - even if they are private. What happened to the ideal of separating interface from implementation? To make matters worse, current C++ tools can't distinguish between changes that affect the object size and those that don't, thereby causing many unnecessary compilations. And make no mistake, C++ compilation can be slow when large numbers of library classes are included.

In short, productivity depends on class libraries and development environments, not on programming languages. Today, there are a few reasonable C++ libraries but no good environments.

C++, by itself, can't help you with improved product quality and easier enhancement either. To get these benefits the system must be designed in an object-oriented style which extracts the best from the language. Object-oriented design is still in its infancy. It cannot be comfortably integrated with other analysis and design approaches but requires, to maximise its benefits, that the whole software development life-cycle be revised. Few CASE tools support object-oriented development.

A major impediment to the construction of high quality C++ systems is that any C programmer can write correct C++. That certainly doesn't mean they can write good C++. Many organizations like the idea of C++ because they believe that its similarity to C will reduce training costs. Experience suggests that such savings are small. Good software is good because it is based on a sound architectural framework. Designing good frameworks with classes is hard and takes years of practice. People with skills in this area are in short supply.

Object-oriented software development can deliver the touted benefits, but not through a language alone. Don't be misled. C++ might be today's most popular object-oriented programming language but the most productive of object-oriented software developers aren't using it.

John Daniels is a Consultant at Object Designers Ltd, an independent software consultancy specialising in object-oriented technologies. He can be contacted on 0279 755396.

We are looking for contributors to this new column. If you have an opinion or idea that you would like to air here, please contact the Editor at the address given on page 1.

ADA COMPILERS

The new AdaStudent is an affordable way to learn this important language.

AdaGraduate	PC-DOS	£625
AdaVantage Dev Kit(R14)	PC-DOS	£1500
AdaVantage Prof Dev Kit	PC-DOS	£2495
AdaStudent	PC-DOS	£ 75
Janus/Ada compiler	PC-DOS	£355
IntegrAda	PC-DOS	£645

PRICES & DELIVERY

Prices do not include VAT or other local taxes but do include delivery in UK and Europe. Please check prices at time of order, ads are prepared some weeks before publication.

This page lists some of our products. Call us for a complete pricelist.

Order by phone with your credit card.

PROGRAMMING TOOLS

Ada Compilers	Algol Compilers
Assemblers & Libs	AWK
Basic Compilers	Basic Interpreters
Basic Utilities	Basic Libraries
BCPL Compilers	C Compilers
C Interpreters	C Libraries
C Utilities	Cobol Compilers
Comms.Libraries	Cross Assemblers
Database Libs.	Debuggers
Dis-assemblers	Editors
Engineers Libs.	Expert Systems
Forth	Fortran Compilers
Fortran Libraries	Graphics Libraries
Icon	Linkers/Locaters
Lisp	Modula-2
Nial Interpreters	OPS 5
Pascal Compilers	Pascal Libraries
Prolog	Rexx
Risc	Screen Libraries
Smalltalk	Snobol

We stock many items for which there is no space in these advertisements.

THE C LANGUAGE

Microsoft C V6 is a complete rewrite with improved optimisation and a new Programmer's Workbench High C V1.6 has been considerably improved, with better Microsoft C compatibility, and new documentation

C COMPILERS

Avocet C	PC-DOS	£210
Aztec C86 Professional	MS-DOS	£120
Aztec C86 Developer	MS-DOS	£175
Aztec C86 Commercial	MS-DOS	£290
CI C86 Plus v1.3	MS-DOS	£290
ECO-C88 V4.0	MS-DOS	£ 70
HIGH C (Metaware) v1.6	MS-DOS	£420
HIGH C 386 v1.6	PL386&MS-DOS	£635
Hi-Tech C	MS-DOS	£135
Lattice C v6	OS/2&MS-DOS	£175
Let's C v4	MS-DOS	£ 60
Microsoft C v6	OS/2&PC-DOS	£269
Microsoft QuickC v2	PC-DOS	£ 60
M'soft QuickC/QuickAsm	PC-DOS	£105
MIX C	MS-DOS	£ 20
MIX Power C & lib s'ce	PC-DOS	£ 29
Turbo C v2	PC-DOS	£ 79
Turbo C++	PC-DOS	£140
Topspeed C for OS/2	OS/2	£335
Topspeed C (std)	PC-DOS	£135
Topspeed C Professional	PC-DOS	£270
Watcom Express C v7	PC-DOS	£ 65
Watcom v7 opt comp + tls	PC-DOS	£250
Watcom C/386 std	PL386&MS-DOS	£625
Watcom C/386 Prof	PL386&MS-DOS	£830
Zortech C v3	PC-DOS	£ 48
Zortech C++ v2	PC-DOS	£120
Z'tech C++ v2 Developer	PC-DOS	£270
Aztec C86 Developer	CP/M-86	£245
Aztec C86 Personal	CP/M-86	£140
Hi-Tech C	CP/M-86	£150
Lattice C	CDOS	£395
HIGH C (Metaware)	Flex OS 286	£420
Avocet C	CP/M-80	£210
Aztec C Personal 1.06D	CP/M-80	£120
Aztec Commercial 1.06D	CP/M-80	£200
Hi-Tech C	Z80 + CP/M-80	£ 99
Mix C	Z80 + CP/M-80	£ 35
Toolworks C/80 v3.1	CP/M-80	£ 45
Aztec C65 V1.05	Apple DOS	£200
Aztec C68K MPW C	MACINTOSH	£110
Aztec C68K C	MACINTOSH	£ 75
Aztec C68K/AM Developer	AMIGA	£175
Aztec C68K/AM Prof'snl	AMIGA	£120
Aztec C68K/ST Developer	ATARI	£175
Aztec C68K/ST Prof'snl	ATARI	£120
Hi-Tech C	ATARI	£ 99
Lattice C v3.04	ATARI	£ 85
Mark Williams C v2	ATARI	£110
Laser C	ATARI	£135
Prospero C	ATARI	£105

C CROSS COMPILERS

We supply 2500AD, Avocet, Aztec, Lattice, IAR and Hi-Tech Cross Compilers hosted on MS-DOS and targeted on Z80, 6502, 6801, 68HC11, 6301, 6809, 7811, 8051, 8096, 68000 & 68020. Please call for information or advice.

C INTERPRETERS

Instant C/16M supports large programs on 286/386 machines.

C-terp V3.0X	PC-DOS	£165
Instant C/16M	PC-DOS	£725
Interactive C	PC-DOS	£195
Introducing C	PC-DOS	£ 85
Living C Plus	PC-DOS	£135
RUN/C Professional	MS-DOS	£110

C LIBRARIES

DATABASE/FILEHANDLING

Btrieve	OS/2	£400
Btrieve v5	MS-DOS	£170
Btrieve/Network	MS-DOS	£400
XQL	OS/2,PC-DOS	£540
Xtrieve Plus	OS/2,PC-DOS	£400
C/Database T'lchest sce	PC-DOS	£ 39
CBTREE	(source any C)	£135
C-Index/Plus	(source any C)	£175
C-ISAM (L,MS)	MS-DOS	£220
C-tree	(source any C)	£230
Essential Btree (s'ce)	PC-DOS	£120
CQL s'ce	PC-DOS	£270
Faircom Toolbox Prof	PC-DOS	£665
Pro-C & Workbench	PC-DOS	£300
R-tree	MS-DOS	£170
db-VISTA III s'gle user	PC-DOS	£575
db-Query v2 s'gle user	PC-DOS	£575
Lattice dBCIII (L,MS)	MS-DOS	£185
Lattice dBCIII+	OS/2&MS-DOS	£365
SftFcus Btree&Isam	(sce any C)	£ 80
Virtual Memory File Man.	PC-DOS	£ 80
Zortech Btree/ISAM	PC-DOS	£ 75

GRAPHICS

Enhanced Graphics Tlkt	PC-DOS	£170
Essential Graphics v3	PC-DOS	£235
GFX Graphics	PC-DOS	£ 75
GraphiC v5 (CI,L,DS,MS)	PC-DOS	£290
Graphics Menu/META	PC-DOS	£135
GSS Kernel system	OS2,PC-DOS	£350
GSS Graph Dev T'kit	OS2,PC-DOS	£350
HALO v3 (MSC5,LAT,TC)	PC-DOS	£230
HGraph (MSC)	PC-DOS	£ 80
Ingraf (MSC)	PC-DOS	£175
MetaWINDOW (CI,L,MS,T)	PC-DOS	£145
MetaWINDOW/Plus	PC-DOS	£190
MetaWINDOW/Premium	PC-DOS	£315
TurboWINDOW/C (Turbo)	PC-DOS	£ 79
QuickWINDOW/C (QC)	PC-DOS	£ 79

SCREEN & WINDOWS

Vitamin C (MS5.1)	OS/2&PC-DOS	£240
Power Screen (MS&TC)	PC-DOS	£105
Blaise View Mngr.(s'ce)	PC-DOS	£245
Lattice Curses	PC-DOS	£105
Entelekon Windows(s'ce)	PC-DOS	£105
Multi-windows (MS,L)	PC-DOS	£190
Panel Plus II (source)	PC-DOS	£270
Vitamin C (source)	PC-DOS	£165
Vermont Views (MSC,TC)	PC-DOS	£345
Greenleaf Datawindow	PC-DOS	£180
Greenleaf Makeform	PC-DOS	£ 60
C-Scape(+s'ce&Look&Feel)	PC-DOS	£330
Zortech Windows(ZC++ZC3)	PC-DOS	£ 39
Curses/PC w. s'ce	PC-DOS	£265

GENERAL FUNCTIONS

C+O Class Lib (MS,TC)	PC-DOS	£155
C+O Cl's Lb (MS)WINDOWS	&PC-DOS	£155
C/Utilities T'chest sce	PC-DOS	£ 39
C-Worthy Int.Lib (L,MS)	MS-DOS	£175
Code Runner	PC-DOS	£105
Greenleaf Functions s'ce	PC-DOS	£135
G'leaf Super Functions	PC-DOS	£175
G'leaf Fin'cial Mathlib	PC-DOS	£230
The Heap Expander(MS&TC)	PC-DOS	£ 65
Smorgasbord C-food	PC-DOS	£ 65
C Tools Plus/6.0 (MS)	PC-DOS	£105
Power Search	PC-DOS	£105
Turbo C Tools v2	PC-DOS	£105
Essential C Utils s'ce	PC-DOS	£145
Entelekon Funct(source)	PC-DOS	£105
Resident-C (MS,L)(s'ce)	PC-DOS	£145
Silverware C EMM Lib	PC-DOS	£115
WKS Library	PC-DOS	£169
Zortech Supertext	PC-DOS	£ 39

COMMS LIBRARIES

C Asynch Manager V3s'ce	PC-DOS	£135
Essential Comms(L,MS,T)	PC-DOS	£195
Greenleaf Comms(source)	PC-DOS	£205
Multi Comms (L,MS)	PC-DOS	£190
Silvercomm C Asynch	PC-DOS	£145
Zortech Comms (QC&TC)	PC-DOS	£ 65
Zortech Comms(ZC++&ZC3)	PC-DOS	£ 65

SCIENTIFIC LIBRARIES

C Lang Sci Lib v2	ANY C	£175
Mathpak 87 (L,MS)	MS-DOS	£ 85

PROGRAMMERS UTILITIES

PC-Lint	OS/2,MS-DOS	£ 85
C-Documentor	PC-DOS	£195
C-Scan	PC-DOS	£195
Clear+ for C	PC-DOS	£165
Lattice Comp Companion	MS-DOS	£ 65

REAL TIME & MULTI-TASKING TOOLS

Concurrent C (PC/MPX)sce	MS-DOS	£ 60
Multi C (L,MS,CI,TC)	PC-DOS	£190
Op.Sys.T'box (MS) s'ce	PC-DOS	£ 65
Timeslicer v3.01 (L)	PC-DOS	£165
Timeslicer v5 (MS)	PC-DOS	£165
Over-C (L,MS)	PC-DOS	£225

ASSEMBLERS

MS Macro-86 v5.1	OS/2&PC-DOS	£ 95
PharLap 386 ASM.LINK	MS-DOS	£405
PharLap 386 ASM/Linkloc	MS-DOS	£965
PharLap 386/VMM	MS-DOS	£240
Optasm	MS-DOS	£ 95
Turbo Assembler/Debugger	PC-DOS	£105
2500AD Z80 ASM	CP/M-80	£ 70
Dig.Res. RMAC	CP/M-80	£180
SLR Z80ASM	CP/M-80	£ 40
SLR Z80ASM-PLUS	CP/M-80	£140
SLR MAC	CP/M-80	£ 40
SLR MAC-PLUS	CP/M-80	£140
SLR 180 (Hitachi)	CP/M-80	£ 40
SLR 180-PLUS (Hitachi)	CP/M-80	£140

Not all assemblers are supplied with a linker. Check before ordering.

DISK COPYING SERVICE

We can copy files to and from 600 disk formats including CP/M, CP/M-86, MS-DOS, PC-DOS, ISIS, APPLE, SIRIUS, BBC, TORCH, APRICOT, HP-150, TRSDOS, DEC RT-11, IBM BEF, ATARI ST, AMSTRAD, MACINTOSH. Our charge is £10.00 + disk + VAT with discounts on small quantities and disks are normally despatched within 24hrs of receipt. For more information call us.

OBJECT ORIENTATED LANGUAGES

The new V2.0 Guidelines C++ is now in stock

C SUPERSETS

Guidelines C++ v2.0 (MS)	PC-DOS	£370
Zortech C++ v2	PC-DOS	£120
C-Talk	PC-DOS	£115
Turbo C++	PC-DOS	£120
Turbo C++ Professional	PC-DOS	£200

PASCAL SUPERSETS

MS Quick Pascal	PC-DOS	£ 60
Turbo Pascal V5.5	PC-DOS	£ 79

SMALLTALK

Smalltalk/V	PC-DOS	£ 85
Smalltalk/V 286	PC-DOS	£155
Smalltalk/V PM	OS/2	£350

OTHERS

Actor v3	Windows3+PC-DOS	£545
RisC	PC-DOS	£ 55

GREY MATTER

2 Prigg Meadow, Ashburton, Devon TQ13 7DF

TEL. (0364) 53499

GREY MATTER

2 Prigg Meadow, Ashburton, Devon TQ13 7DF

TEL. (0364) 53499

GREY MATTER

2 Prigg Meadow, Ashburton, Devon TQ13 7DF

TEL. (0364) 53499

News

PDQ

PDQ is a library for Microsoft QuickBASIC V4.0 or later and BASIC compiler V6.0 or later. It replaces the default library supplied by Microsoft. Code modules linked to it form executables as small and fast as those produced by C and Pascal (it says here). There is also a facility to produce TSRs. PDQ V2.0 has just been released; it is available direct from its US creators Crescent Software (0101 203 438 5300) priced $129 + $35 shipping.

Doctor DOS

I thought that Digital Research's DR DOS V5.0 was already shipping, but a man dressed up as a stork alerted me to the fact that it was not by coming round to our offices and 'laying' a copy of the just-released package. How amusing. DR DOS is also more expensive than I previously reported - it costs £119.95, from Megatech (081 874 6511).

Code Base upgrade

Code Base V4.2 is the just-released updated version of the Code Base dBASE library, which is reviewed elsewhere in this issue. The library now includes a browse/edit facility, and can be used under MS-DOS, OS/2, UNIX or Windows. The MS-DOS and OS/2 version costs £190 ex VAT, and a 'portability' version is £320. The distributor is the Software Construction Company (0763 244114).

Lahey FORTRAN

Lahey is a US company specialising in FORTRAN. It has a 32-bit MS-DOS compiler called F77L-EM/32, which requires the company's DOS Extender product, called Ergo, in order to work. This compiler has just reached Version 3.0 - new features include an editor and make utility. US prices are $895 for the compiler plus $395 for the Extender. Lahey has also upgraded its small memory model 16-bit 'Personal FORTRAN', which costs US $99. UK distributors are Grey Matter (0364 53499) and System Science (071 833 8222).

Software Users Year Book

The Software Users Year Book is a four volume catalogue of all the commercial software available. Whether you want to find out how many 32-bit LISP compilers are available for your machine, or need to find an accounting and stock control package suitable for a small pharmacy in Basingstoke, you will find it in here. The 1991 edition has just been published, it costs £130 + £5.50 p&p. It is available direct from its publishers: VNU (071 439 4242).

OS/2 V1.3?

Forget OS/2 V2.0, with fully-fledged 80386 support. There is now a very real danger that the next released version of OS/2 will be 16-bit. IBM recently dropped hints about a revised version of OS/2 V1.2, which could get by with about 3 MB of RAM and ditto of disk space. The existing effort requires at least an extra megabyte of each. Other enhancements are to include a better 8514 display driver (5 bits per pixel instead of just two) and the bundling of Adobe's Type Manager, which contains fonts superior to those currently offered, resulting in faster screen handling.

As is traditional, IBM declined to name a date when the product might appear, although it did indicate that work on reducing the memory requirements of the OS was well in hand. With pre-release versions of OS/2 V2.0 already being tested by selected developers, one would imagine that V1.3 must appear within a few months if it is to make any impact.

Don't compare us with Clipper

Force is a dBASE compiler which its distributors, Continuous Technology Ltd, do not want compared with Clipper. In order to foster this spirit of non-comparision, the company has sent me a set of speed benchmarks which compares Force with... Clipper (very favourably, natch).

A telephone call provided enlightenment. Force achieves its high speed and small executables by throwing out the sections of the dBASE language that are slow and bulky to implement; for example macro handling. By the same fix, it also avoids having to worry about memory fragmentation. The designers have instead introduced various C language features, such as a preprocessor and a proper, stack-based calling convention that lets you hook into standard C libraries (although the library supplied with the kit sounds quite reasonable - it is a complete reimplementation of Turbo C's). You can even do TSRs. The flip side of the coin is that your dBASE applications will require significant modification before they can be compiled under Force - for example, Force requires that variables and file structures are all explicitly declared before use. It is this weakness that makes comparison with Clipper specious.

Force costs £695 ex VAT, Continuous Technology Ltd is on 0372 727269.

An APL Interpreter- and change from £10

The I-APL project exists to promote the use of the APL programming language, especially in education. To this end, it has produced an APL interpreter which runs on a range of small machines such as the BBC Master, Research Machines Ltd's Nimbus plus the inevitable PC-compatible; ports to the Mac, Amstrad 8256/8512 and Commodore 64 are in the pipeline. Copies of the interpreter are sold at cost price; sharing with friends is encouraged, although you are not supposed to make use of the software in commercial applications. Not that you are likely to do this: I-APL is built on a 64 KB memory model, and the interpreter leaves only 30 KB for the user.

If you fancy having a go with this unusual language, then a starter kit (PC version on 360 KB disk + a 55 page manual) costs £5.50, including P&P. Cheques should be made payable to I-APL Ltd, and orders sent to 56 The Crescent, Milton, Weston super Mare, Avon BS22 8DU. Enquiries should be directed to Anthony Camacho, 2 Blenheim Rd, St Albans, Herts AL1 4NR.

UNIX - The Legend Evolves

As this issue of .EXE went to press, the UK UNIX systems User Group was holding its Summer 1990 Technical Conference at the Royal Lancaster Hotel in London. This week-long event, which commenced on the 9th July with two days of tutorials, attracted some 420 UNIX devotees. Some very well-known industry luminaries were speakers.

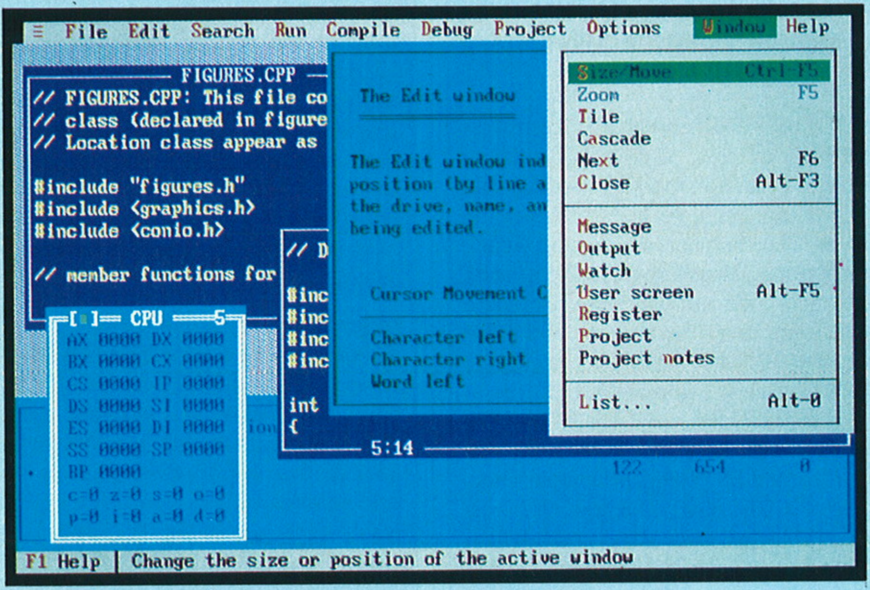
On the first full day of the conference, the sessions covered: Plan 9, a UNIX successor currently under development at AT&T Bell Labs; C (papers presented by Dennis Ritchie and Ken Thompson) and C++ (a passionately argued paper by an ex-Bell Labs staffer, Tom Cargill, who is trying to see Multiple Inheritance removed from the forthcoming ANSI C++ standard); networking and distribution protocols. Later papers covered UNIX internals, advanced graphics and music applications, software productivity (presented by Brian Kernighan) and distributed systems.

Your reporter was intrigued to see several conference attendees wearing T-shirts depicting our hero Bjarne Stroustrup with a large wood screw through his head. We would like to track down the source of these T-shirts - anybody who knows where they come from please contact .EXE's news desk. There's a .EXE T-shirt in it for the first person to produce the information.

For details on the UKUUG call 0763 73039.

Paul G Smith

HotShot



This the story of how I solved one of .EXE's problems; but it might be one of yours as well. When we review software, we often throw in a few screen shots, to give you a better idea of what the package is like. Usually these are text mode screens - editors, debuggers and the like. To get these I have had to send a diskette off to a photographic studio where they took photographs of a PC's monitor. The results were unevenly exposed and, thanks to the curvature of the CRT, never quite in focus.

The crisp, sharp picture of Turbo C++ that you are looking at (or, if you are not, bear in mind that these pages were assembled on Friday 13th) was obtained with a package called HotShot Graphics. HotShot can grab a screen in most common video modes, including text and VGA graphics. There are special modules to handle Windows and OS/2 PM. It has some rather neat editing facilities, so you can doctor the captured screens. But the best part, in the current context, is the facility to convert screens captured in text mode into the bit image format required by laser scanner machines. The photo was produced from a PCX file, but HotShot does about half-a-dozen other common formats. If your manual or brochure does not run to colour printing, HotShot also produces grey-scale images, suitable for importing into Ventura, WordPerfect etc.

HotShot Graphics costs £179.95, and is available from Ctrl Alt Deli (0908 662759). Our picture was produced from a PCX file by Advanced Business Graphics (081 569 9944), which charges £12.50 per slide.

MultiBase

One of the large number of dBASE-related announcements this month concerned MultiBase. This is a MS-DOS based multi-user database which contains its own multi-tasking executive - it is a bit like a copy of Concurrent DOS and a database bundled together. The multi-tasking is done in protected mode, so you will need an AT or better. Up to 30 terminals can be attached via serial ports - a free bundled comms program allows PCs to be set up as dumb terminals - or, if this is not fast enough, an ARCNET version is available at extra cost. The language is near dBASE III+ compatible, supporting many Clipper extensions.

The most obvious drawbacks to MultiBase: the data file format is non-standard, so data has to be converted explicitly to/from dBASE; and you need to distribute a run-time module with your applications, which is licensed at a cool £499 a shot. The development system itself costs £599, add £100 for ARCNET support. The UK distributor is Rede Products Ltd (0793 513855).

QuickC brought up to date

Microsoft has announced the availability of version 2.5 of QuickC and version 2.51 of QuickC-with-QuickAssembler, bringing these products in line with the recently released C Professional Development System V6.0. QuickC now incorporates the C 6.0 language extensions - based pointers, and `#pragmas` to force parameter passing in registers - plus improved ANSI conformance, a tiny memory model for .COM files and better data access and structure browsing. QuickAssembler remains essentially unchanged - you still need MASM V5.1 to do 286/386 work. Plain QuickC retails at £75, QuickC-with-QuickAssembler costs £125.

OS-9

OS-9 is a 680x0-based real time operating system which supports some UNIX-like features. The primary novelty of a new release, V2.3, is 'coloured memory' which allows designers the flexibility of assigning processes to different memory types; presumably fast, frequent processes get the static RAM. Microware UK, the manufacturer, is on 0489 886699.

20 MB in your pocket

Citizen - the company that makes the watches - has announced a 3.5" floppy drive that can read and write 20 MB of data per disk. What makes this feat especially impressive is that the drive can also cope with conventional 3.5" disks formatted to 720 KB and 1.4 MB. The drive has a SCSI interface, and will start shipping to US OEMs within a few months.

Ovum backs GUI

Ovum Ltd (071 255 2670) is a consultancy which periodically publishes expensive but thorough reports on IT industry trends. The latest, entitled 'The Business Desktop: the Next Five Years' (£660) predicts that the GUI will finally take off in the next few years, and that, in the future, vendors of desktop software will strain hard to make their products compatible with each other's.

Windows 3 SDK

Microsoft has shipped the Windows 3 Software Development Kit. The package costs £400, or £100 for an upgrade from a previous version. The kit includes a copy of CodeView for Windows, resource editing tools, optimisation tools and new source-code examples. You do not need to upgrade to C V6.0 to use it, C V5.1 is sufficient. A separate kit is available for developing device drivers.

POP++

Integral Solutions Limited (0256 882028) has combined its POP-11 AI language with the LISP Flavours system to produce an object oriented version of the language. No prizes for guessing what it is called. POP++ fits into the company's POPLOG development environment, and is available on Sun, DECstation, HP9000 and Apollo workstations, plus certain minicomputers.

Bye bye Lattice

Sad news, this. Lattice is transferring all responsibility for its C compilers (MS-DOS, OS/2, Amiga and various cross-compilers) to its parent company, SAS Institute Inc. By September, there will no longer be a Lattice C. Lattice will continue to look after its RPG II compiler, C libraries and various application software.

Serious compilers

I thought of MicroWay as a company that distributed frosty (but pricey) add-ons to make your PC go faster: transputers, interesting coprocessors, that sort of thing. It does indeed supply this sort of hardware, but it also has a line in 32-bit compilers, and this range has just been expanded and upgraded.

The NDP-486 FORTRAN, Pascal and (dual-dialect BSD UNIX and ANSI) C compilers use the same technique as Phar Lap's DOS extender to allow 32-bit applications to cohabit with MS-DOS. In fact, you can use them with Phar Lap's toolkit if you like - but you do not need it, and you do not have to pay any run-time license money on applications produced with the NDP software alone. The compilers support the Weitek 4167 floating-point coprocessor (if you have not got one, MicroWay can sell you one), and use special global optimising techniques to map code and data onto paragraph boundaries. This helps speed things up because it improves the chances of the 486 loading the whole of a code loop into its on-board RAM cache - enabling it to execute the loop without ever having to fetch instructions from slow external RAM.

Programs produced with these compilers are *seriously* fast, says MicroWay. The Whetstone single precision benchmark, compiled with NDP-486 FORTRAN, and running on a 25 MHz 486 equipped with a 4167, recorded 11.75 MegaWhetstones/second - faster than the floating point performance of a DEC 3100. The compilers cost £750 each. If you only run to a 386 machine (but you will need some sort of coprocessor, if only an 80287), then you might care to know that there is a parallel range of NDP-386 compilers, which has just been upgraded to V3.0. These cost £545 apiece. Further details from MicroWay, which can be reached on 081 541 5466.

Remote control PC

Co/Session is one of those packages that lets you run a remote PC from your own keyboard via a comms link; it feeds your keystrokes into the slave's keyboard buffer and echoes changes to the remote screen on your own. This is brilliant for providing technical support, via a phone line and a pair of modems, without having to go out on site. Special features of Co/Session include: graphics support up to VGA, a proprietary sliding window error-correcting comms protocol and a command processor (optional) which can perform scheduled tasks automatically. The basic pack costs £199.95, the add-in Script processor another £99.95, from Megatech (081 874 6511).

OOP Prolog Standard

Quintec Systems Ltd, noting the alarming proliferation of Prolog implementations with OOP extensions, has suggested that the software houses involved should adopt an industry standard of syntax for implementing objects. Integral Solutions' POP++, LPA's Prolog++ and Quintec's own Quintec-Objects all offer similar features: encapsulation, class inheritance etc. Quintec says that its system has been well received from a technical standpoint, but potential customers have expressed concern that there is no agreed standard for OOP extensions.

If Quintec has any success with its proposal, perhaps it might care to pass on its wisdom to Borland, Microsoft and JPI. Borland and Microsoft have already created two incompatible sets of Pascal OOP extensions; JPI may produce yet another when TopSpeed Pascal appears later this year.

SSADM for dBASE

The phrase 'Structured Systems Analysis and Design Methodology' (SSADM) conjures up images of government contracts, huge mainframes, vast software projects... Checkpoint Computers Ltd of Berkshire says it has tamed SSADM, reducing it to a concise set of manuals that it would like to sell to you for £99 (inc VAT and P&P). 'Rapid Database Standards', as the manual set is called, is targeted at the dBASE user (like nearly everything else this month). According to Mr Jim Hackett, MD of Checkpoint, the company has been able 'to reduce the jargon prevalent in the mainframe standards' and offer practical tips 'which highlight errors before they become expensive and embarrassing.'

The 'Rapid Database Standards' set of manuals is available now. Checkpoint can be reached on 0488 71467.

Oracle to Clipper

Biton (rhymes with 'right on') is a Clipper (Summer '87, but a Clipper 5 version promised soon) library that provides connectivity with the Oracle RDBMS. The idea is that you use Clipper to throw together an elegant front-end, but retain Oracle for the heavy database work. There are side benefits as well; for example, you can send you calculations down to Oracle, it can do them in 62 digit precision arithmetic.

Biton is principally aimed at corporates who wish to get PC programmers up to speed with the company database without having to retrain them. It costs £400, and comes direct from the manufacturer, a company also called Biton (0727 50658). .EXE

dBASE Graphics

If you use dBASE, Clipper, Foxbase, Quicksilver, Force, dBFAST, Vulcan or R:BASE, you may be interested in the latest version of the dGE business graphics package (exploding pie charts etc). Version 4 includes Supertext with variable font size, new type faces and an editor, file-based icons, a driver for colour dot matrix printers, mouse functions and a Norton Guide. It costs £245 from Bits Per Second Ltd (0273 727119).

Index race

dB Race creates new indexes for dBASE compatible data files. Because these indexes are smaller, it is claimed that it can perform sorts and searches in as little as one tenth of the time of the conventional software. dB Race, which is sold as an add-on for dBASE III+ and clones, costs $399 from its US creator LoadStone (0101 704 568 8095).

It's Baby Ada

Milspec Systems can do you a PC Ada compiler for £99 ex VAT. The drawback is that Personal Ada V2.0 only lets you have 64 KB of code. Based on the last time I witnessed an Ada compiler in action, I would estimate that this restricts you to the 'Hello World' program, only leaving out the 'World' - but I dare say Ada has come on a long way since then. The proper, full-sized compiler costs £359, and there's also a new graphics library, priced £99. Milspec is on 0203 670770.

3 for II

*APL*Plus II Version 3 is the latest release of a 32-bit version of APL from STSC International. It contains its own MS-DOS extender, so it can move its interpreter and new session manager into extended memory and switch the CPU into protected mode. You can then load DOS TSRs, such as network drivers, 'underneath' the APL without interfering with it. Other new features include an interface to Lotus 1-2-3, and mixed language programming with FORTRAN. STSC's number is 0753 831451.*

Intel cross compiler

Microtec Research (0256 57551) has announced MCC86 V2.0, an Intel 8086/186/286 optimising C cross compiler which it says is fully ANSI conformant. Hosted on Sun workstations and PCs, the compiler produces re-entrant ROM-able code for the above-mentioned CPUs. There is also target support for the NEC V20-V35 range of processors, and a degree of Microsoft compatibility - for example, near and far pointers are supported. The object code is in Intel MSC-86 object module format.

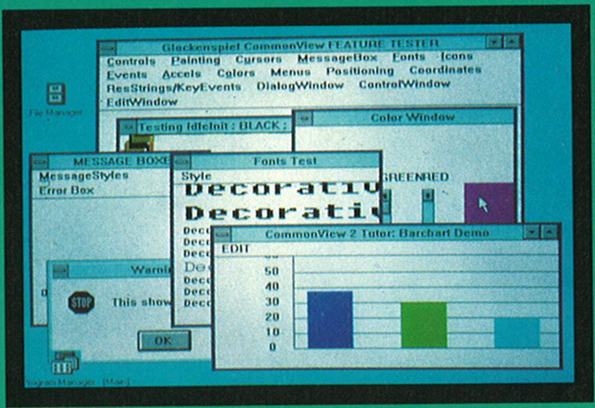
Objects of Desire

glockenspiel CommonView 2

***Glockenspiel CommonView** has really made its mark in the world of Windows development. Thousands of developers have used it to speed up and simplify their projects.*

Now **CommonView 2** is available, inheriting the success of its predecessor and extending its capabilities even further to deliver efficient Windows 3.0 apps.

That's because **CommonView 2** works with **Glockenspiel C++ 2.0**, giving you a C++ object-based framework that reduces the complexity, cuts the code, manages memory and lets you stay in touch with what you're really doing. From compilation to execution, **CommonView 2** applications are fast and powerful.



CommonView 2 objects model the real world of Windows. And that's just the beginning - from there **CommonView 2** takes you as far as you want to go.

It doesn't anticipate everything. That's where C++ takes over: its inheritance capabilities enable you to extend the **CommonView 2** framework. Just take an existing object and customize it without affecting the original. A few additional lines of code and you could, for example, turn an edit control into a password control. Then reuse it in other applications.

You *don't* have to learn a complete new language, you *can* integrate existing developments; it *is* portable to PM and NewWave; CodeView *can* be used on your C++ source.

The world wants Windows 3.0 apps now! And with **CommonView 2** you can inherit the earth.

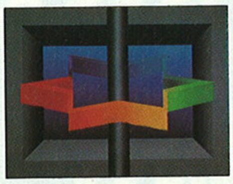
Specifications.
Glockenspiel CommonView 2 includes Glockenspiel C++ 2.0 and Container - the object storage framework. It requires Microsoft C 6.0, the Windows SDK and 1.5 meg of memory. You debug C++ source with Microsoft CodeView 3.0. Glockenspiel C++ supports a completely portable memory management system. Glockenspiel CommonView consists of approximately 65 classes.

Comprehensive documentation includes CommonView tutorial and reference manual, Glockenspiel C++ compiler manual and User Guide, C++ syntax and AT&T Library Guide, pullout guide to compiler switches, *plus* "Programming in C++" by Stephen C. Dewhurst and Kathy T. Stark (Prentice Hall).

On-line hypertext documentation for CommonView reference manual and AT&T guides. Tutorial source code also on disk.

Glockenspiel C++ works from within the Programmer's Workbench with the reference guides instantly available from the on-line Advisor, using Microsoft CodeView for debugging.

Glockenspiel CommonView applications are portable between Windows 2.1 and Windows 3.0, PM 1.1 and PM 1.2 with HP New Wave 3.0 version coming soon.



glockenspiel
class constructors

QA

QA Training Limited,
Cecily Hill Castle,
Cirencester, GL7 2EF.
Phone: 0285 655888. Fax: 0285 650537.

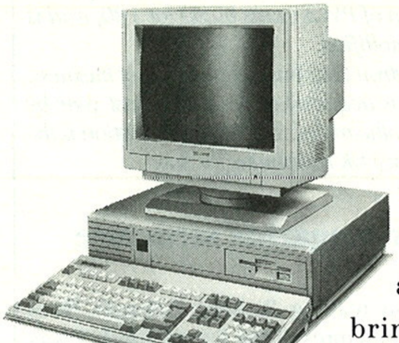
CommonView costs £495 + VAT and is available for Windows 3, OS/2 Presentation Manager and HP NewWave.

Glockenspiel Professional C++ 2.0 Compilers are available for DOS, OS/2 and Workstation platforms. Call for details.

Glockenspiel CommonView and Glockenspiel C++ are registered trademarks of Glockenspiel Ltd. The trademarks of their respective corporations are acknowledged.

CIRCLE NO. 127

Tandon launch the fastest 486 computer under £5,000.



Which is more mind bending: the speed of our new PCAsl/486 or the £4,999 price tag?

Let's look at the speed first. For starters, there's an Intel 80486 processor that crunches numbers like they're going out of fashion.

(Tests have indicated that the 486sl runs at a break-neck 10.3 m.i.p.s.)

Just as hair-raising is the 486's built-in maths co-processor (it tots up figures faster than a bookie on Derby Day). And our unique Power Poster feature, a high speed cacheing system, sprints into action so quickly it'll bring tears to your eyes.

However, £4,999 doesn't only buy you sheer unadulterated speed. There's a cavernous 110MB of hard disk storage capacity and the high clarity VGA colour monitor makes the most humble pie charts look tasty.

And don't forget that the 486sl can use not only existing PC compatible software, but also the latest 32-bit graphical interfaces such as Windows 3.

Take heed though, at under £5,000 we predict the 486sls won't be hanging around for long.

Tandon

For more information on the Tandon PCAsl/486 write to Tandon plc, FREEPOST, Hunt End, Redditch, Worcester B97 5XP. Or call us on (0527) 550550.

Name.. *Position*.. EXEJ/486

Company.. *Address*..

....

Telephone.. *Number of employees*........ *Number of PC's installed*........

Prices exclude VAT. 80486 is a registered trademark of the Intel Corporation. Windows 3 is a registered trademark of the Microsoft Corporation.

CIRCLE NO. 128

Letters

We welcome short letters on any subject that is of interest to our readers. Please write to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. Unless your letter is marked 'Not for Publication', it will be considered for inclusion on this page.

Polytron saga

Sir,

I am enclosing correspondence *(not reproduced - Ed)* that I have had with Polytron concerning the use of their version control system. As you can see from the letters I sent them, the PVCS system has been corrupting files - an unforgivable feature of any program, but particularly of a version control system! The letter (with no originating address) recently received from a software engineer acknowledges that the product had a bug of this sort and recommends me to upgrade to version 3.3.

Fair enough. I telephoned Grey Matter and they referred me to the UK distributor for upgrades - a firm called Software Generation. A very helpful person there said that since I had had the product for more than 12 months, I would have to purchase the product again - at a cost of £495. At this point I complained and pointed out that a) the product I had purchased had failed to work as advertised and b) I was a registered user of this (and many other) Polytron products and was not provided with any update information at any time.

The helpful person said that this was unfortunate but couldn't be helped since Polytron themselves had gone out of business and basically I would have to start over again. (I then understood why the software engineer in the US did not write back on Polytron business paper - or indeed from any organisation at all).

So I have been had. I purchased a program which failed to do what it was advertised to do - worse, it actually did the opposite! And now I have been told that the only way to rectify the situation is to fork out another £500. Readers of .EXE beware. Polytron PVCS corrupts files under a number of circumstances - and you'll have to pay for the privilege of finding out whether the new product does the same!

Peter Chapman
Energy Advisory Services
Hanslope

This story has a happy ending. We sent a copy of this letter to Software Generation and, as a result, the company experienced a change of heart. Prof Chapman has been offered an upgrade to the latest corporate version of PVCS (sans bugs) for £80, and is now mollified.

Polytron itself has not gone out of business, but was acquired at the end of last year by Sage Software Inc. Software Generation is the exclusive UK distributor for Sage.

Poor magazine structure

Sir,

Upon reading Richard Pickard's 'Program Structures of the Fourth Kind', I was amused to find that the article read in the order pages 27, 34, 30, 33 and finally, 28. Was this accidental or was it some sort of sick joke on your part?

Steve Dix
Dowty Magnetics
Staffs

Sir,

In your July issue, the excellent article 'Program Structures of the Fourth Kind' seemed to be missing three things: Labelling (page numbers on some of the pages), GOTOs (to tell you which page to go to next) and Block Structure (pages started mid sentence!). Luckily I managed to find the correct sequence by using the figure numbers.

Can we have page numbers on every page please - and prompts such as 'continued on page ...'; and surely in these days of computerised typesetting, we don't need to split sentences between pages!

Clive Craske
ROCC Training
Crawley

Many apologies to Richard Pickard, whose article was messed up. If you abandoned an attempt to read it as a result of the foul-up, please do go back and give it another shot, as it will certainly repay your effort.

This was a rare-ish example of a genuine printer's error (two half-pages of editorial were exchanged), as opposed to far more common goofs such as typing er1234rors, wrong picture captions etc, which originate in this office. I have forwarded Mr Craske's suggestions to She Who Controls The Layout, and her answers are: numbers on pages - sorry, no can do, would cause hassle with our advertisers; sentences not split across pages - not convinced this is necessary; 'continued' - good idea, she had been advocating it for yonks, watch this space.

Sparks from May

Sir,

If Jules May (Soapbox, .EXE July 1990) is such a good programmer that he has produced only one bug in 10 years of programming, why does his choice of programming language matter?

G M Wright
Wormley
Surrey

Sir,

Well done to Jules May! It was about time that somebody said it!

C J Glick
London

Contented Fractint

Sir,

First, many thanks indeed for getting a copy of Fractint to me within four days of my sending you a disk. That's a performance that most software companies and shareware libraries couldn't match!

Unfortunately, I happen to be one of those unfortunates who do not own a modem, so I'd be grateful if you could pass on my thanks to the authors of the program. I've had a couple of earlier versions and all of them have given me more pleasure than any arcade game. I'm not a C programmer, so I can't contribute to the project directly, but I would like to suggest one useful add-on for a future version; support for colour printers and (especially) plotters. HPGL would be preferable. I generally use a camera for hard copy, but it would be nice to get something really accurate, without the inevitable distortion of a curved screen etc. It would be especially useful with the 3D images.

Marcus L Rowland
St Marylebone School
London

EXE

An object lesson in programming

C++

Meet Your Objective

Object-Oriented Programming (OOP) is programming for the '90s. It's the next step after structured programming and is a much more productive way of writing applications. Borland has combined the power of OOP with the efficiency of C to produce new Turbo C++ Professional®.

Turbo C++ Professional is the first Turbo-charged native code C++ compiler that brings Object-Oriented Programming to your PC. Since Turbo C++ Professional also compiles ANSI C code, you can be productive with C now, and move to C++ at your own pace.

Environment ++

The best compiler deserves the best environment and our new Programmer's Platform® turbo charges your productivity. It features overlapping windows and mouse support, as well as a new multi-file editor, an integrated debugger, and a smart project manager. Its advanced open architecture lets you integrate the tools you need and are familiar with.

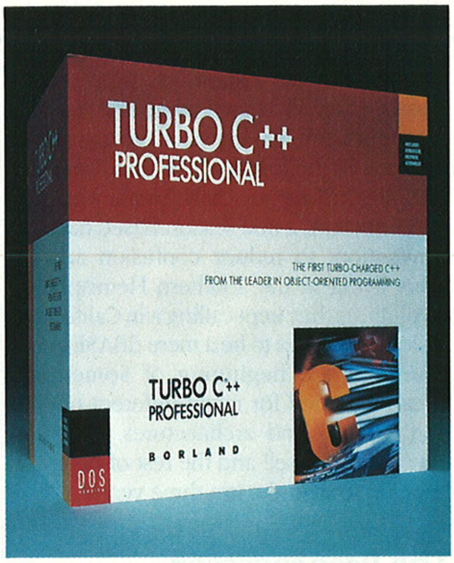
VROOMM adds room

VROOMM® (Virtual Run-time Object-Oriented Memory Manager) lets you break through the 640K barrier. Just select the application code you want to overlay, and VROOMM does the rest – swapping modules on demand. It's fast, easy, and automatic.

Another +

Turbo C++ Professional gives you all the tools you need to build fast, reliable C++ programs.

Turbo Debugger® 2.0 debugs your object-oriented programs. This powerful new version is the first and only debugger to support reverse execution, letting you step backwards through your code to find the bugs



you might have missed.

New Turbo Profiler®, the world's first interactive profiler, gives detailed analysis of your program's performance. With it, you can easily spot execution bottlenecks and see where improvements or redesign of your code will yield maximum performance gains. And Turbo Assembler® 2.0 lets you replace time-critical segments of your code using the world's fastest MASM® compatible assembler.

Buy objectively

The suggested retail price for Turbo C++ Professional, including Turbo Debugger, Turbo Assembler and Turbo Profiler is £249.95 plus VAT. Turbo C++ costs £149.95 plus VAT. For further information complete the coupon and return or call our PRODUCT INFOLINE on (0628) 771070 or simply talk to your dealer.

Turbo C++ Compiler

- C++ conforming to AT&T's 2.0 specification
- C++ class libraries
- Full ANSI C compiler
- VROOMM overlay manager
- Complete documentation and tutorials

Programmer's Platform

- Open architecture for integration of your own tools
- Overlapping windows with mouse support
- Multi-file, macro-based editor
- Smart project manager provides visual MAKE
- Integrated debugging and hypertext help

YES! Please send me a free Turbo C++ demo disk. ❐ $3^{1/2}$" or ❐ $5^{1/4}$"

NAME: ______________________

POSITION: ______________________

COMPANY: ______________________

ADDRESS: ______________________

______________ POSTCODE: ______________

EXE 7/90

TELEPHONE: ______________________

Please return this coupon to: Borland International (UK) Limited, FREEPOST RG1571, Twyford, Berks RG10 8BR

CIRCLE NO. 129

BORLAND

All Borland products are trademarks or registered trademarks of Borland International. Other brand and product names are trademarks or registered trademarks of their respective holders. Copyright © 1990 Borland International Inc.

Not so much a compiler, more a statement of direction

Clipper V5.0 is ending its career as vapourware. Ian Butterworth has been taking a look at the latest dBASE-ish compiler from Nantucket.

dBASE World

Welcome to the first thematic issue of .EXE Magazine. Although entitled 'dBASE World', we have ended up with a heavy Clipper bias. Perhaps this is partly a reflection of Ashton Tate's current difficulties; and perhaps also that, because it is a compiler, Clipper is the Developer's (as opposed to the Lay User's) favourite.

The names on the continents and satellites in our cover illustration were produced by going round the office, inviting everybody to name all the PC-based DBMSs that they could remember. Omitted manufacturers should therefore address complaints to their own marketing people, and not to me. *WRW.*

The original dBASE compiler was written by Brian Russell back in 1984. He took it to Ashton Tate, the dBASE manufacturer, but the company wasn't very interested. So one evening Brian Russell met one Barry ReBell in a restaurant called *The Nantucket Light.* The upshot of this occasion was the formation of the Nantucket Corporation, with ReBell as Chairman. The company's product was Russell's compiler, which was launched as Clipper Summer '85.

Since that time, Clipper has grown to be pretty successful, with over 200,000 users worldwide. One consequence of this success is that Clipper has been able to create its own (dBASE-incompatible) *de facto* standard, and has spawned a whole community of developers of both end-user applications and add-on products.

The current (pre V5.0) Clipper release is Summer '87. At the US Nantucket Developers' Conference in California, which I attended a few weeks ago, we were given 'The Conference Edition of Clipper 5.0'. This is, in effect, a gamma release version of the software; it is this package that forms the basis of the following review. The new compiler marks a radical break with the past - not least a non season-based naming convention, to reduce confusion among users living in the Southern Hemisphere. Five-Oh, as they kept calling it in California, no longer aspires to be a mere dBASE compiler, but the beginning of Something Great, intended for many different operating systems and architectures. So what have Brian Russell and the rest of his team been up to over the last three years?

The Preprocessor

The facilities offered by the new preprocessor will be boringly familiar to C programmers - but very exciting to dBASE language developers. There are directives `#define` and `#undef`, for the creation (or deletion) of manifest constants, pseudo-functions and so on; `#include` to incorporate header files (containing all those `#defines`), `#ifdef`, `#ifndef`, `#else` and `#endif` to allow conditional compilation.

Two other directives, geared specifically to Clipper syntax, are less familiar. `#command` turns what appears as a command in the source file into a function call. Many of the existing Clipper commands have been reimplemented this way, and it provides a powerful mechanism for substituting other database engines, screen drivers etc into the same source code. `#command` allows a variable number of parameters, the ability to wrap parameters in quotes (for example, as in the `USE` statement, which normally takes a literal file name) without changing any of the source. `#translate` expands four-letter abbreviations of commands (a hangover from interpreted days) into the full version, but otherwise behaves just like `#command`.

Scoping and Data Types

Another hangover from the interpreter is Clipper's weak set of scoping rules. If I declare (ie use) a variable name within a procedure, any routine that it calls can 'see' that variable. If a procedure with a loop `FOR i=1 TO n` calls another routine with a loop `FOR i=1 TO n`, the caller's variable `I` is changed by the inner one! Until now, the only way to prevent this happening was to declare all local data as private - which left the responsibility of hiding data to the called and not the caller.

There are two new variable declarations: `LOCAL` and `STATIC`. `LOCAL` means that the variable is only visible in the routine in which it is declared, and it is stored on the stack. This appears, on the surface, to be a minor point, but I believe it will make Clipper code generally more robust.

`STATIC` is like a C `static` - its value is retained between invocations of a routine. For convenience, `STATICs` can have file-wide scope (when compiled with the /N switch, a module's `STATICs` are visible to all its routines). `STATIC` is a very useful mechanism for data hiding. Previously, one had to declare a large number of `PUBLIC`

Clipper V5.0 new features

- Preprocessor
- Better scoping and new data types
- Source-level debugger
- Database independence
- New operators and functions
- A new dynamic linker

variables which could then be trashed by any careless piece of coding. Now, the details of libraries can remain safely hidden from applications.

LOCALs and STATICs have a number of other features in common: for one thing their names are resolved at compile time. PRIVATE PUBLIC and, a new one, 'FIELD' variable names are known at runtime, which means they can be used in macro substitution. The snag is that each has an overhead of 22 bytes per variable name. LOCALs and STATICs lose this memory overhead, plus one level of indirection, so we should be able to look forward to better performance in accessing memory variables.

There are also such things as STATIC functions, which are visible only in the module in which they're defined. Once again, this serves to enhance the hiding of program and data. Henceforth there should be less use of undocumented (and therefore changeable at Nantucket's whim) Clipper functions. For those who don't see what is wrong with this sort of thing, consider how many Microsoft Windows V2.X developers now wish that they had used memory as they were told?

Another aspect of the generality of Five-Oh is the introduction of code blocks, which can be thought of as pointers to functions. Essentially, you can assign a chunk of code to a variable at compile time (thus avoiding the overhead of runtime macro expansion). This chunk of code can be passed around just like any variable and executed using the function eval(). Here is a code block which takes the parameter yy, passes it to func and then increments it:

```
xxblock := {|yy| func(yy), yy++}
```

Code blocks are often used in internal functions. By default, asort() sorts an array into ascending order. By passing the code block parameters, you can force descending order or specify behaviour for dealing with two different test data types.

Arrays have been expanded to structures of potentially quite labyrinthine complexity - not like arrays in any other language that I have seen. Previously, array elements could be of different types, perhaps reflecting the structure of whole database files. This has been expanded to allow arrays to contain references to other arrays, so we have the possibility of an unlimited number of dimensions (except for memory limitations). Arrays can now be initialised when they are declared:

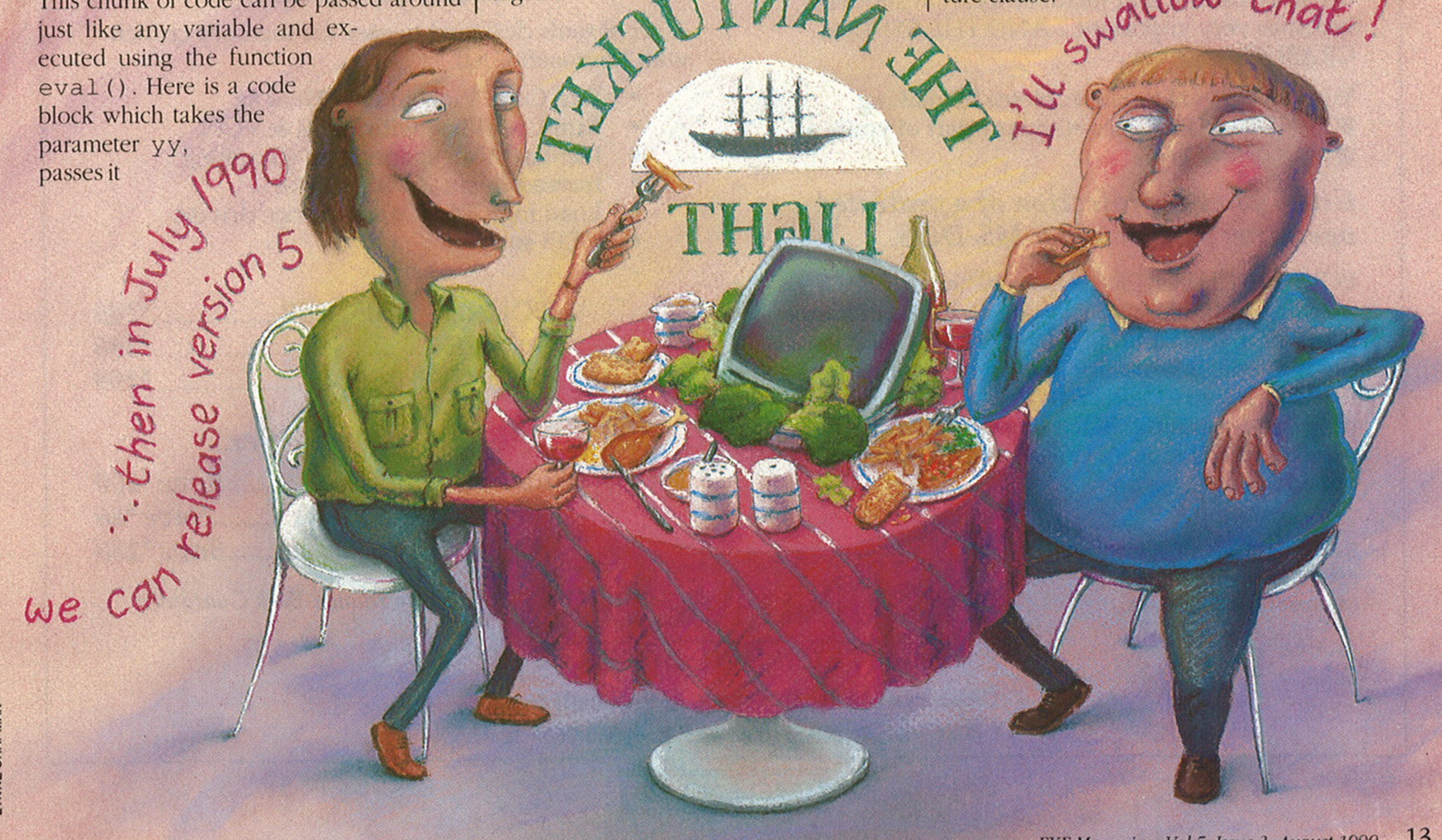
```
LOCAL nice_lads := {"Smed",;
        "Malc", "Mark", "Skye",;
                  {38, 25, 32, 47}}.
```

Some functions to handle the new arrays are shown in Figure 1. You can now declare LOCAL parameters in a style reminiscent of FORTRAN: function AtSayCol(nRow, nCol, cText, nColour). These parameters are all 'genuine' ie actually passed on the stack, with scalars passed by value, multiples by reference. For these purposes, a string counts as a scalar. Parameters may be skipped in function invocations by the simple expedient of placing two commas together - the called routine is passed the new constant NIL. FIELD is the last new data type and is used to define memory variable names for use with a particular database. It is most useful to the programmer who duplicates field names all over the place (forbidden practice in my company).

GETting better

One outstanding weakness of the current system of obtaining input is the inability to nest GETs. For example, if you are entering data against an order, and suddenly discover that your database doesn't have the required code for some new product, the only way around it is to abort the read, go to the product entry screen, type in the new product data and then start all over again from scratch.

Five-Oh's GET system solves this problem. You are now allowed to have multiple GET lists on the go. You can either suspend the current GET list and drop down to an inner level (nested GETs), or, for the ambitious, you are allowed to have more than one list running simultaneously (parallel GETs). Other new features include: full program control over navigation and editing keys, cargo information associated with each GET allowing (eg help text to be displayed when a user is entering data in a particular field); the availability of state information (insert key status and so on); and dynamically configurable state information such as colour and picture clause.



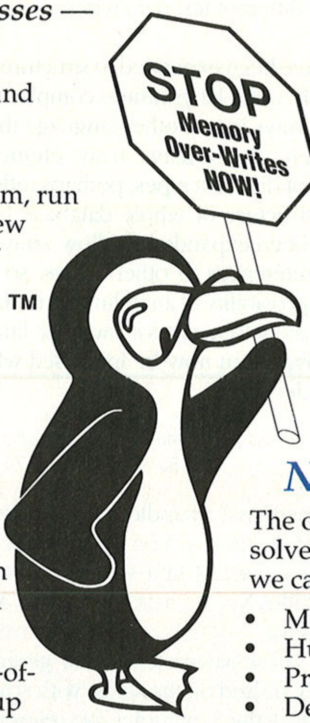
POWER DEBUGGING

BOUNDS-CHECKER

Finds out-of-bounds memory accesses — AUTOMATICALLY.

Flush out those Nasty pointer problems and other out-of-bounds memory accesses — AUTOMATICALLY.

Each time you make a change to a program, run BOUNDS-CHECKER while testing the new code. If you accidentally access out-of-bounds memory, BOUNDS-CHECKER will pop up displaying the offending SOURCE LINE. And your program runs at full speed.



"BOUNDS-CHECKER and Soft-ICE make sophisticated use of the most powerful versions of Intel's processor family to track down some of DOS programming's most insidious bugs. If you're developing programs for DOS, these are essential tools."

PC Magazine
July, 1990 pg. 48

Ship Bug-Free Products

You can run BOUNDS-CHECKER while testing your program. There are no additional steps to your testing cycle, but you can feel secure when the program has passed through BOUNDS-CHECKER with no reported problems.

Many over-write problems and other out-of-bounds memory accesses do NOT show up during normal testing. An out-of-bounds memory location may be modified, but that particular location doesn't happen to be important at the time. Once the program is in the field and a certain network is loaded or a certain T&SR or device driver is loaded, that memory location suddenly becomes very important... AND THE SYSTEM CRASHES.

You can prevent these problems by making BOUNDS-CHECKER a standard part of your testing procedure.

Gives you the protection of a protected operating system under MS-DOS.

BOUNDS-CHECKER uses the 80386 virtual machine technology to provide real-time memory protection. In addition BOUNDS-CHECKER uses the symbolic information output by your compiler to differentiate CODE and DATA. When your program is running, BOUNDS-CHECKER protects the program's CODE and all memory outside your program.

Requires 80386 PC.
MS-DOS is a trademark of Microsoft Corporation.

Soft-ICE 2.5
New Version, New Features

The only debugger specifically designed to solve those problems unique to MS-DOS that we call the DOS Nasties.

- Memory over-writes
- Hung programs
- Program too big to debug
- Debugging T&SRs and Loadable Drivers
- Multiple Symbol Tables
- Supports Microsoft C 6.0 & Turbo C++

MagicCV 3.0
New for C6.0

A set of tools designed to ease the memory crunch with Microsoft C 6.0.

- Run Code View in ***Less Than 8k***
- Run CodeView with EMM & VCPI
- Increase heap space when compiling
- Increase memory with make
- Load high T&SR's and device drivers
- VCPI support

BOUNDS-CHECKER *$249*
Soft-ICE 2.5 .... *$386*
MagicCV 3.0 .... *$199*

Special Offer...

Buy BC & S-ICE .... ***Save $100***
Buy S-ICE & MCV .... ***Save $86***
Buy all three .... ***Save $186***

30 Day Money-Back Guarantee

Nu-Mega
TECHNOLOGIES

CALL TODAY (603) 888-2386 or FAX (603) 888-2465
P.O. BOX 7607 ■ NASHUA, NH ■ 03060-7607 ■ U.S.A.

CIRCLE NO. 130

Figure 1 - Operators and Array Functions.

Operators

Operator	Example	Meaning
:=	x := y	x = y and return the value y to LHS
+=	x += y	x = x + y if numeric or concatenate if character
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
++	x++	x = x + 1 also in prefix form
--	x--	x = x - 1
^= or **=	x ^= y	x = x ^ y Exponential
%=	x %= y	x = x MOD y

Array Handling Functions

Function	Example	Meaning
aclone()	x := aclone(y)	Make an exact copy of this array
aadd()	aadd(x, y)	Add element y to array x
asize()	asize(x,5)	Make first dimension of x 5
acopy()	acopy(y,x)	Copy all elements of y to x
aeval()	aeval(x, y)	Execute codeblock y on each element of x

The other array functions appear to work in a similar fashion to Summer '87.

The revised system is based on things called 'GET objects' - which, as a confirmed OOP-o-phobe, makes me nervous. The preprocessor translates each `@...GET` command into an object which contains characteristics of the `GET` command itself. Each `GET` is added to an array called `GetList`. `GetList` is `LOCAL` to the routine in which it is defined, so nested `READ`s become a natural part of the way the language works, and not a bolted-on work-around. When a READ command is issued, the preprocessor translates it into a call to the standard function `readmodal(GetList)`. The default method of operation is as in Summer '87.

The values in `GET` objects are set and retrieved using code blocks which are evaluated at runtime. It is possible to assign a new value to the `GET` variable or just retrieve its value. The `GET` object needs to know practically nothing about the variable being 'GOT'. Validation can now be carried out before, during and after the variable is being GOT. For example, a VALID clause is now translated to be a code block executed after variable entry.

I have not yet had time to investigate all the ramifications of this new approach, but it does not take much imagination to see how one might implement a multiple window application where the user can switch between activities at will - suspending some and activating others according to need. The new `GET` code itself has been written in Clipper - the source file GETSYS.PRG is supplied - so it is very user configurable.

The Browse System

The new Browse system (replacing the slightly strange `dbedit()`, although `dbedit()` is still available in this release) is implemented in a similar style to `GET's`. As the reference manual says, 'A TBrowse object is a general purpose browsing mechanism for table-oriented data'. Data retrieval and record positioning is implemented via user-supplied code blocks passed as messages to the TBrowse object in question.

A TBrowse object is created using the function `TBrowseNew()` which defines the name of the object and its screen size. New columns (ie data fields) are created by passing the `AddColumn` message to the browse object - please see Figure 2. Each column or block of cells may be individually coloured, so you can easily highlight queried data or negative values. Colours are also set by passing messages.

Stabilisation is the process whereby the screen is redisplayed after movement within the database(s) being browsed. It is often convenient to be able to prevent/abandon this process (for example when the user is paging down rapidly) so you do not waste time redrawing the screen when the next keystroke, waiting in the keyboard buffer, will force another refresh. Five-Oh lets you do just this, so the user can move to the next screen of data without having to wait for the current screen to fill up (which may take some time, if complicated look-ups are involved).

Although TBrowse and `GET`s act like objects, the manual is at some pains to say that Five-Oh is NOT an object-oriented language.

Database Independence

Although I touched on database independence in the context of user defined commands, a little more should be said about it. By including non-standard header files to, it is possible to reference another set of functions to perform data retrieval. This should mean that the lonely freelancer, toiling in his garret, can develop with a dBASE III+ compatible file structure, then, once it is ready for delivery, recompile using a non-standard header file and, lo and behold, it's Netware SQL or whatever the customer wants.

Drivers should be available both from Nantucket and third parties over the next few months. The following drivers have been mooted: Paradox, Microsoft/Sybase SQL Server, Gupta Technologies SQLbase, Novell's Netware SQL and Oracle RDBMS. Note there are Summer '87 drivers available for Oracle and SQLbase.

```
Function BROWSEDIR( aBrowse, nTop, nLeft, nBottom, nRight , cMatch)
Local oBrowse, oColumn, nElement := 1, lCont := TRUE, I, nKey

oBrowse := TBrowseNew( nTop, nLeft, nBottom, nRight )

oBrowse:goTopBlock  := {|| nElement := 1 }
oBrowse:goBottomBlock := {|| nElement := Len( aBrowse[1] )}
oBrowse:skipBlock   := {|nMove| SkipArray( nMove, aBrowse, @nElement )}

oBrowse:colorSpec := "N/W, B/W, B/BG, R/W, W+/R, GR/BR ,+GR/B"

oColumn := TBColumnNew( "File Name", {|| aBrowse[1,nElement]} )
oColumn:width := 13

SetColor("N+/N");@ nTop ,nLeft + 1 Clear To nBottom + 2, nRight + 3
SetColor("B/W");@ nTop - 1,nLeft - 1, nBottom + 1, nRight + 1 Box BOX_SINGLE

Do While (lCont)
//Stabilise browse (until done, or keystroke hit)
Do While (! oBrowse:stable) .And. ((nKey := InKey()) == 0)
oBrowse:Stabilize()
EndDo

//...Everything stabilised
If oBrowse:stable
If nKey == 0
nKey := InKey(0)
EndIf
EndIf

//...Handle keystrokes
Do Case
Case nKey == K_DOWN
oBrowse:Down()
Case nKey == K_UP
oBrowse:Up()
Case nKey == K_ESC
lCont := FALSE
EndCase
EndDo

Return (NIL)
```

Figure 2 - Manipulating a TBrowse Object.

Figure 3 - An RTLink command file.

```
#                  Dynamically overlay an application and use a shared
#                  library

DYNAMIC            # Turn on dynamic overlaying
INCREMENTAL        # Link only changed modules
PLLNEEDED qb51ib   # Link to shared library
FILE mainprog      # User program file
#                  No need to mention Clipper.lib
# Many of the defaults can be set up using the RTLINKCMD environment variable.
```

Miscellaneous Goodies

The assignment operator := returns a value C-style, so multiple assignments can be expressed on one line, as in `x := y := 0`. See Figure 1 for a list of the other new operators. The macro operator can now be used on expressions; before it used only to work on simple variables - not even array elements. The syntax is `&(str1 + str2)`.

Runtime Environment

Clipper 5.0 comes with a simple program editor (actually itself a Clipper program), DBU (a utility program), a source level debugger and a new linker. The editor is pretty basic, and not likely to win many friends. DBU provides facilities for the creation and modification of DBF files.

The debugger is a full screen effort, with one small window for entering commands plus a menu accessed via ALT-key combinations. The main debugger window is given over to the source code being executed, with a highlight on the current line. To set break points, you move the cursor to the line in question and press F7. I have heard rumours that there will be an option to run EGA or VGA screens in 43/50 line mode, with the application screen occupying half and the debugger the remainder - let's hope this is right.

The linker is a vast improvement on PLINK86 - it is faster and it helps to get round one of the bugbears of all Clipper programs - size. The design of the Clipper language requires that much of the Clipper library has to be loaded with each application. This means that a 'Hello World' program in Clipper is around 160 KB! The new linker, a special version of RTLink from PocketSoft, provides several two features to enhance both memory and disc usage: dynamic overlaying and shared library files. In order to speed up the link process incremental linking is available - only changed modules are relinked into the application. Because this special version of RTLink knows about Clipper, basic linking can be carried out with the simple command `rtlink <objfile>`. Response files, such as shown in Figure 3, can be used for more complicated link sessions.

The runtime environment takes a more setting up than before, now that we have include files, dynamic link libraries *et al.* Five-Oh now installs itself into a directory CLIPPER5 with five sub-directories BIN, INCLUDE, LIB, PLL and SOURCE. Fortunately the environment can be set up à la Microsoft, and both the compiler and the linker know about SET INCLUDE, LIB etc.

The compiler itself now takes a greater number of optional parameters, see Figure 4. The package comes with a Norton Guides engine and database, to enable hotkey look-ups which are context-sensitive to the item under the cursor in your editor. A Make utility is also supplied.

Performance

Caveat: this is only pre-release software. Performance was a little disappointing; about 15% slower to compile, producing .EXE files about 10% larger - although this is probably inevitable when supporting dynamic linking. Compared to using PLink86, linking was about three times faster. Running applications, we found that indexing (a traditionally heavy-duty number) was about 15% slower. Nantucket has promised that the released compiler (and its resulting executable) will perform at least as well as Summer '87.

What's the same?

As far as we can tell, Five-Oh is completely source code compatible with Summer '87. An existing application (which only used the Clipper and Extend libraries) had *exactly* the same bugs when recompiled.

In the manual, commands and functions which are obsolescent are marked with an asterisk. We have due warning about what is to be dropped. Generally, the functions marked for deletion are those which are hangovers from interpreted dBASE (eg `CALL`, `TEXT`).

There has not been a vast increase in the number of core functions - if you want graphics, or need to use the mouse, then you'll still have to buy an add-on package. The third party developers have had beta copies of Clipper for some months, so most add-on libraries should be ready shortly after the release of Clipper 5.

In Conclusion

So what has the Clipper development team been doing these last three years? Well, they appear to have spent about 18 months reading a lot of C manuals. They seem to have thought pretty hard about how developing ought to be. I think that they have created an open and very usable compiler. I, for one, am keener to get my hands on the full release than before. Oh? So I like it then? Yeah!

The question that remains is 'When is it actually coming out?'. I think the answer is that we should start seeing it in the UK around the end of July.

EXE

Switch	Meaning
/A	Automatic declaration of PUBLIC & PRIVATE
/B	Include debugging information
/D<ident=text>	Define an identifier
/I<path>	Additional search path
/L	Suppress line numbers
/M	Compile single module only
/N	Suppress automatic main procedure
/O<objfile>	Specify output object file
/P	Produce preprocessor listing
/Q	Quiet mode
/R<libfile>	Embed a library in addition to Clipper.lib
/S	Syntax check only
/T<path>	Path for temporary files
/U<user header>	User's standard header file
/V	Treat all variables as memory (not field)
/W	Generate warning for ambiguous variables

These switches can be set up in the environment using the command: SET CLIPPERCMD=

Figure 4 - Compiler Switches.

Ian Butterworth is a director of Software House and Distributors Quin Butterworth Spangenthal Ltd. Like many ageing programmers, he started off using punched cards on mainframes and ended up working mostly on PCs. Although he still likes to keep his hand in with FORTRAN and COBOL, these days he spends most of his time writing applications in Clipper.

QBS distributes the Blinker linker, reviewed elsewhere in this issue.

POLYTRON

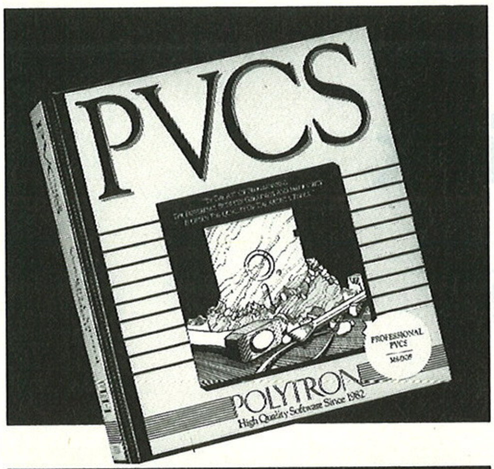
SOFTWARE DEVELOPMENT TOOLS

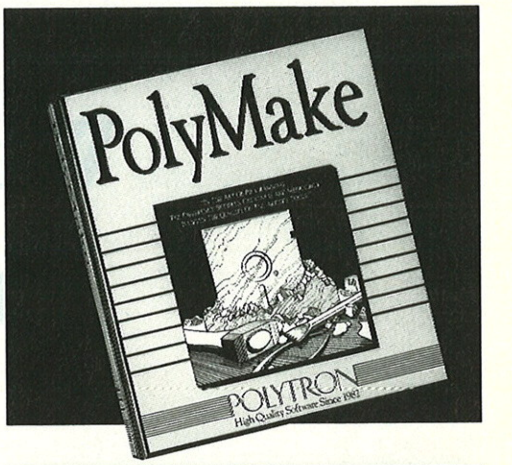
FROM

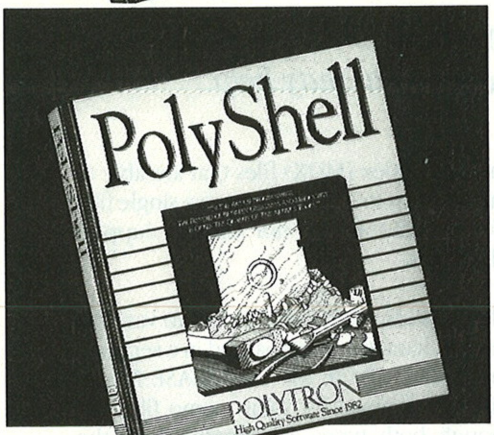
SOFTWARE GENERATION

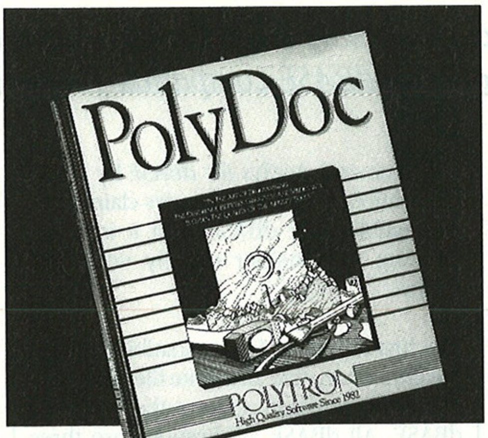
SOLE DISTRIBUTOR

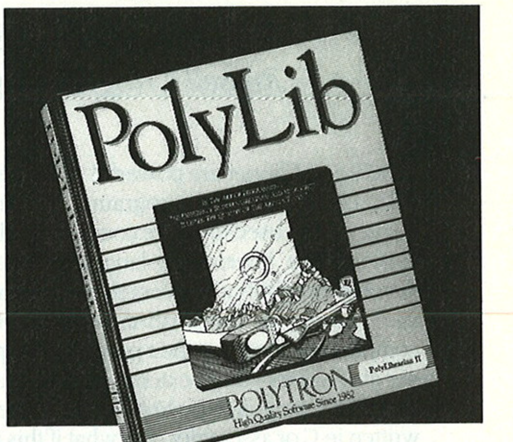
0462 422525

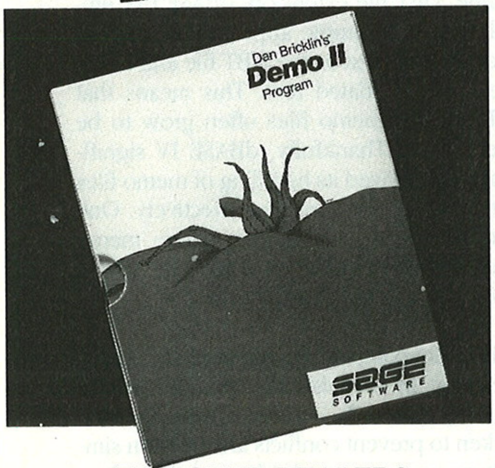


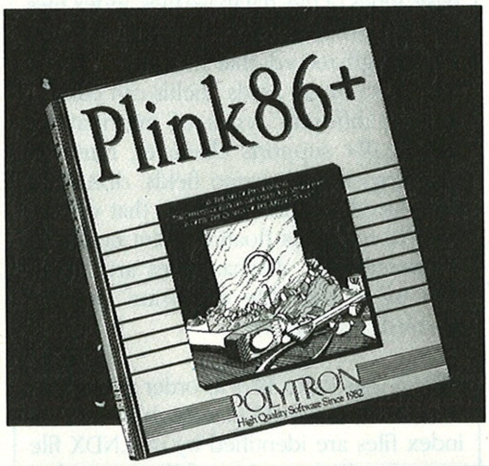


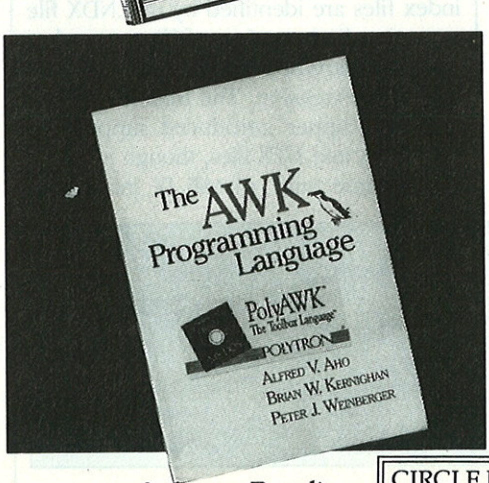












APPROVED DEALERS

Roundhill Computer Systems Limited
Orchard House
Ogbourne St. George, Marlborough
Wiltshire SN8 1SU
Tel: (0672) 84535

Computer Solutions Limited
Canada Road, Byfleet
Surrey KT14 7HQ
Tel: (09323) 52744

Grey Matter Limited
Prigg Meadow, Ashburton
Devon TQ13 7DF
Tel: (0364) 53499

Software Paradise
Avenue House King Edward Avenue
Caerphilly Mid Glamorgan CF8 1HE
Tel: (0222) 887521

CIRCLE NO. 131

APPROVED DEALERS

Frontline Distribution Limited
Intec 1, Wade Road, Basingstoke
Hampshire RG24 0NE
Tel: (0256) 463344

Reflex Technology Limited
9 Buckingham Place
Bellfield Road, High Wycombe
Bucks HP13 5HW
Tel: (0494) 465907

System Science
3-5 Cynthia Street
London N1 9JF
Tel: 071-833 1022

Where the Quick meets the Easy

Do you need dBASE compatibility but feel constrained by dBASE or Clipper? Russell Craigie looks at three dBASE-compatible file managers that may help.

dBASE and Clipper are powerful packages. dBASE's interpreted programming language can be used to create complex applications. The interpreted language eases development but can be very slow, so many use Clipper, or other dBASE-compatible compilers to speed up their programs. Clipper also allows you to extend the power of the dBASE language by linking in routines written in C or assembler. But what if this is not enough? What if you need to maintain dBASE compatibility without using dBASE or Clipper? What if you would like to access dBASE files using the C programming language?

This article looks at three function libraries that allow C programs to access files that are compatible with the files produced by dBASE III+ and dBASE IV. You retain the full power of the C language (for example, allowing the use of low-level code that accepts data from some exotic peripheral) and, at the same time, can reach dBASE databases. Even if dBASE compatibility is not a requirement, the three libraries can be regarded as fully functional Indexed Sequential Access Method (ISAM) file managers.

The libraries reviewed are: *dBC III Plus* by Lattice Inc, *Code Base 4* by Sequiter Software Inc and *AccSys for dBASE* by Copia International. All three libraries claim to be compatible with dBASE, and it is worth pausing to consider exactly what this means.

The first condition for compatibility is that the library can read and write files that have the same internal structure as those used by dBASE. All dBASE applications use three basic types of file: database files, index files and memo files. Database files contain fixed length records that are further divided into fixed length fields. Fields can contain several different types of information; dBASE III+ supports character, numeric, date, logical and memo fields. dBASE IV introduced a sixth field type that can be used to represent floating point numbers. Conventionally, database files are identified by a .DBF file extension and are referred to as DBF files.

dBASE uses index files to order and access the information held in a database. dBASE index files are identified by the .NDX file extension. Early versions of Clipper used an incompatible type of index file identified by the .NTX extension. The Summer '87 release of Clipper introduced support for both NDX and NTX files, though not both at the same time. dBASE IV introduced multiple index (MDX) files that are able to maintain up to 47 index tags in a single file. Neither dBASE III+ nor Clipper support MDX files.

Memo fields contain references to variable length character strings stored in a separate memo file. dBASE III+ and dBASE IV use different formats for their memo files, although both types are identified by the same .DBT file extension. dBASE III+ employs a very simple approach to memos, adding new text to the DBT file and never releasing outdated text. This means that dBASE III+ memo files often grow to be very large. Thankfully, dBASE IV significantly improved its handling of memo files and space is now reused effectively. One implication of this is that, where memo fields are involved, dBASE III+ and dBASE IV are not entirely compatible.

The other compatibility prerequisite is only relevant in a multi-user or multi-tasking environment. It involves the approach taken to prevent conflicts arising from simultaneous access to database records. Provided that the library adequately supports file and record locking, compatibility is only an issue if you plan to use library functions and dBASE simultaneously to access the same files. Programming multi-user applications for networks can be difficult at the best of times - mixing file managers demands particular care.

	dBASE III+	dBASE IV	Clipper (Summer '87)	Codebase 4	dBC III Plus	AccSys for dBASE
Max no of recs in d/base file	1 billion	1 billion	1 billion	2 billion	1 billion	1 billion
Max record size (bytes)	4,000	4,000	65,536	32,666	4,000	4,000
Max no of fields in record	128	255	1,024	1,022	128	255
Max size of character field	254	254	65,535	254*	254	254

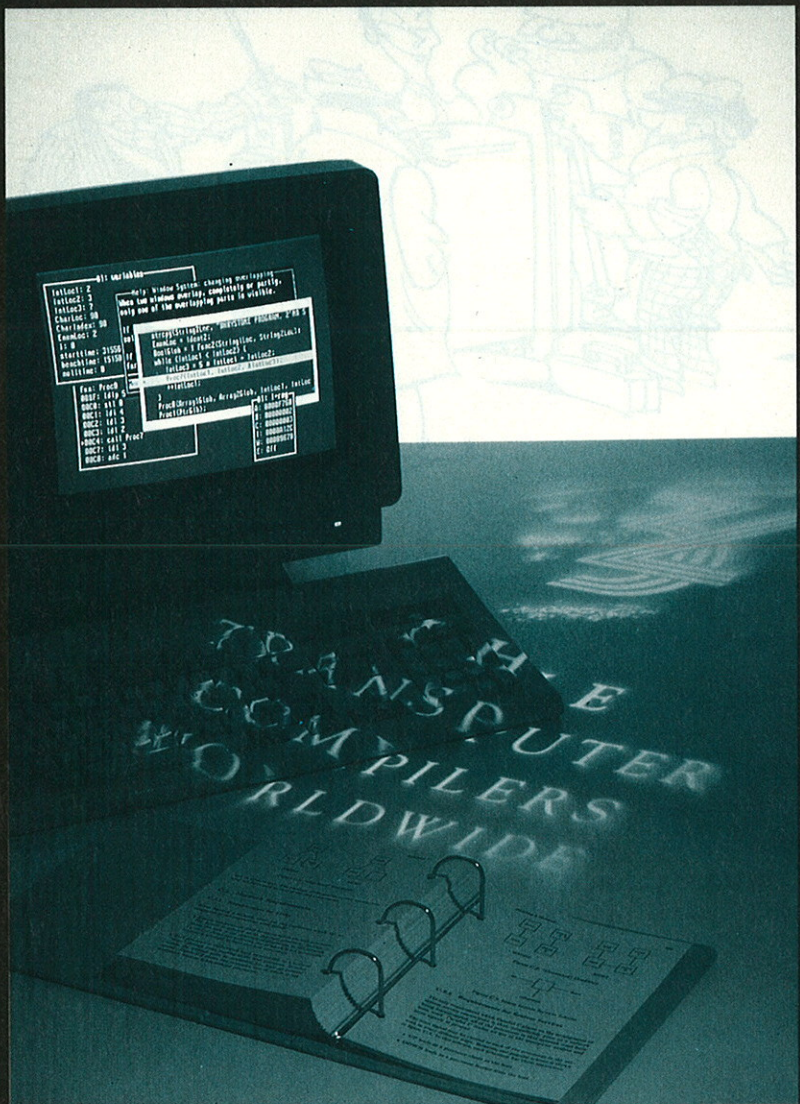
* *for dBASE compatibility. Greater for Clipper.*

Figure 1 - Product Specifications.

Common features

All three libraries implement a core of common functions: they can all create database and index files, open files, read, write and close database and memo files. All provide file and record locking mechanisms, that

THE TRANSPUTER COMPILERS WORLDWIDE



For more information,
ask your transputer hardware supplier,
or write directly to:

3L Ltd.
Peel House
Ladywell
Livingston
EH54 6AG
Scotland

Tel 0506 415959
Fax 0506 415944

IN INDUSTRY
IN EDUCATION
IN FINANCE

In thousands of sites where transputers are being applied today.

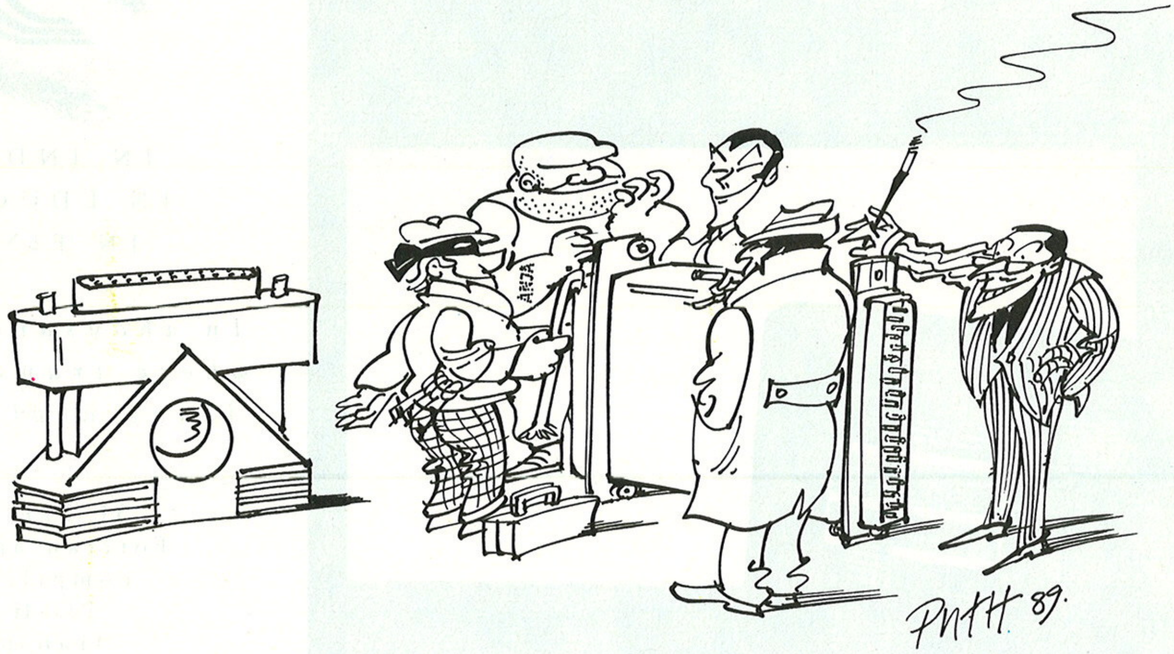
3L Parallel C, Fortran and Pascal compilers give PC-Hosted transputers transparent access to DOS files and graphics-from the DOS prompt, without special languages or operating systems. Developers benefit from the Tbug interactive debugger. And end users see transputer power in a familiar environment.

Of course, choosing the industry standard also gives you the widest range of compatible 3rd party products. Training courses, assemblers, Rowley Modula-2, libraries for graphics, Microsoft Windows, NAG functions, vector and image processing.

PARALLEL PROCESSING
MADE EASY

CIRCLE NO. 132

5 out of 5 hackers prefer other software protection methods to Hardlock E-Y-E®

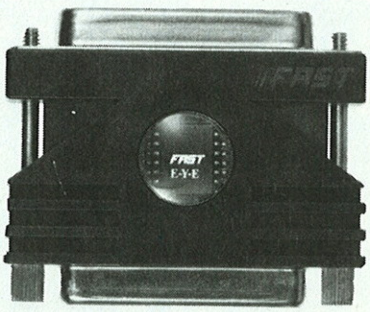


What hackers dislike...
Hardlock E-Y-E was designed using cryptographic principles. It took the experience and know-how of Germany's No. 1 in software protection and the leading edge technology of a US semiconductor company to create the ultimate software protection tool. Hardlock E-Y-E is based on a custom chip featuring secure algorithmic response rather than simple bit swapping or counting schemes.

What software developers like...
Hardlock E-Y-E combines all the features software developers require in a single product: algorithmic response to provide security and an optional non-volatile memory to allow custom configurations. FAST Electronic has made implementation of Hardlock E-Y-E in your software easy. Use HL-Crypt to protect .EXE or .COM files, or incorporate high level language interface routines in your software. The algorithm parameters and the contents of the memory can be programmed in seconds using our Crypto-Programmer card. This unique card guarantees that no one else can burn your original codes. Simply plug the card into any PC slot and start up your own Hardlock E-Y-E workshop.

What your customers will like...
Hardlock E-Y-E allows unlimited backup copies. The device is shipped with the software for the user simply to plug into the parallel interface and forget. Daisy chainability, outstanding reliability (no battery is needed), and the most compact High-Tech design ensure that your customer will accept Hardlock E-Y-E.

What your accountant will like...
Hardlock E-Y-E needs no factory coding. This ensures optimum delivery schedules and stock flexibility. Revenues will go up as software piratry and multiple usage are prevented. Despite its wealth of features, Hardlock E-Y-E's prices remain competitive.



Hardlock E-Y-E
programmable, algorithmic response and memory option – all in one.

...As more and more software developers, customers and accountants appreciate the Hardlock E-Y-E device, hackers like it less and less.

MAGNIFEYE

Order your demo unit today. Contact Magnifeye,
235-239 Walmer Road, Walmer Studio #6, W11 4EY, Telephone 071 221 8024, Fax 071 792 3449.

CIRCLE NO. 133

enable you to write multi-user applications. The libraries use a different approach to accessing fields than the one supplied in the dBASE language. Under dBASE, one must select a record and then explicitly read each field as required. The libraries work by selecting a record and then reading the entire record into a buffer held in memory; individual fields may then be extracted or manipulated as necessary. Writing data follows the same principle - the memory image of the record is updated and the entire record is then written back to the database.

dBASE and C use different conventions for representing data. Strings, numbers and dates are all stored in the database as fixed length character strings. Unlike C, dBASE has no need for a null character to mark the end of the string. To overcome these, all three libraries provide a set of conversion routines that allow you to switch from one representation to the other.

Lattice dBC III Plus

dBC III Plus is a library of 61 functions compatible with files used by (surprise, surprise) dBASE III+. Users of Lattice's C compiler will already be familiar with this library, since it is the multi-user version of the dBC III library bundled with the Lattice C Development System for DOS and OS/2. dBC III Plus is sold as a separate product in its own right and is available in versions that support Microsoft and Turbo C compilers, as well as Lattice's own.

dBC III Plus's set of functions to manipulate database, index and memo files provide both power and flexibility. Note that it is possible to exploit this flexibility to create database files that are not compatible with dBASE III+, although the manual provides very explicit guidance on how to ensure that compatibility is maintained.

dBASE III+ compatibility results in a number of limitations that are not shared by the other libraries reviewed here. First, records may contain a maximum of 128 fields, compared with 255 for AccSys and 1,022 for Code Base 4. Second, memos created using dBC III Plus are limited to 4,096 characters. The library does contain a variable that specifies the maximum size of a memo field but the documentation does not specify the range of acceptable values. dBC III Plus does not support dBASE IV memo or multiple index files. Figure 1 summarises the technical specifications of the three libraries.

dBC III Plus implements a full set of file and record locking functions, and you can use it to write an OS/2 Presentation Manager application that accesses a dBASE III+ database. It is supplied with a 212 page ring-bound manual, organised into six chapters followed by appendices containing a detailed description of each function and a list of error codes. Separate chapters provide an overview of the product, a set of design guidelines, a sample application, a discussion of dBASE III+ compatibility and guidance on using dBC III Plus under OS/2. The manual is very clear and easy to read.

AccSys is the only library that supports dBASEIV's Multiple Index files

The basic price for dBC III Plus includes only the object code for the library; modules for small, compact, medium and large memory models are provided. You must specify which manufacturer's compiler you require when you order the package. If you wish to use any other compiler you will need to purchase the source code direct from Lattice.

Code Base 4

Code Base 4 (henceforth CB4) is a library of database and screen management functions that are compatible with the data files used by dBASE III+, dBASE IV and Clipper. CB4 is produced by Sequiter Software Inc of Alberta, Canada. Unlike the other libraries reviewed, CB4 provides a set of screen management functions that allow you to create windows, display menus and read data using functions similar to dBASE's `GET` routines. This additional functionality means that the combination of CB4 and a C compiler can be regarded as a full replacement for dBASE and Clipper.

CB4's database functions all correspond to real dBASE commands. They allow the programmer to create and maintain database, index and memo files. The library supports both the dBASE NDX and Clipper NTX formats for index files; there is no support for dBASE IV's multiple index files. Records have a maximum size of 32,666 bytes and may contain up to 1,022 fields. CB4 allows you to perform the standard operations to define and build index files in addition to providing access to lower-level routines.

There are two distinct sets of functions for accessing memo files. The first supports the memo file format used by dBASE III, dBASE III + and Clipper, the second is compatible with the memo file format used by dBASE IV. Both provide the same functionality. The key difference between them is that the dBASE IV routines will recover disk space when writing memo file information. There is also a call that allows you to convert dBASE III memo files into dBASE IV format. Memos may contain up to 32,767 characters.

CB4 also provides a memo edit routine that works by copying the memo to an external file, invoking a user specified external text editor to edit the file, and then copying the file back into the memo field. CB4 does not implement the word-processing features of Clipper's `MEMOEDIT` function, nor does it attempt to provide any of the supporting functions such as `MEMOLINE` and `MLCOUNT` (neither do the other libraries). Programmers moving from Clipper to C will miss this functionality, but C loyalists will be pleased with the simplicity and turn to their favourite libraries for help.

`GET` commands are emulated. Data entry fields are placed using a series of get 'commands' and activated by issuing a read 'command'. The operator may tab backwards and forwards between fields thus created, and there are facilities for picture templates and simple validation of entered data. Some set of pre- and post-modifier routines allow you to specify the attributes, associate a message and define a template for an area. Finally, CB4 also supports pop-up, pull-down, vertical, horizontal and Lotus style menus. I doubt that many C programmers will be tempted to abandon their favourite I/O libraries to use CB4's. However, these facilities will be handy for dBASE/Clipper aficionados making the transition.

CB4 implements a set of file and record locking functions for use in an OS/2 or network environment. CB4 uses a record locking strategy that is compatible with Clipper.

The standard package contains all the library source code. The bulk of it is written in C, with one assembly language module. The library explicitly supports Microsoft C, QuickC and Turbo C and contains pre-compiled library modules for each of these compilers. The source code contains several compilation `#define` switches that allow you to build libraries that are compatible with OS/2 and Windows, and that use Clipper NTX files. Another removes all refer-

ences to the screen management routines, so you can use a separate I/O library if required. Finally, there is a switch that allows you to create applications using very small executable files - provided that you do not use index files.

CB4 is supplied on four 5.25" diskettes, with a 213 page spiral-bound manual. The manual is clear, but should do more to present an overview of programming using CB4. In addition to the library source and object code, several example programs are included. No royalty payments are required when you distribute DOS or OS/2 executable programs that incorporate CB4 routines. A version of CB4 is available for UNIX/XENIX environments.

AccSys for dBASE

AccSys for dBASE contains 120 functions, compatible with both dBASE III+ and dBASE IV. It is the only library reviewed here that supports dBASE IV's Multiple Index (MDX) files. AccSys for dBASE is produced by Copia International, and is part of a family of libraries designed to permit access to the files used by popular database management programs. AccSys for dBASE is available in both single-user and multi-user forms; I looked at V1.0 of the multi-user version.

I doubt that many C programmers will be tempted to abandon their favourite I/O libraries to use CB4's

AccSys contains an extensive set of functions for manipulating database and index files. Databases created using AccSys have specifications identical to dBASE IV: each database may contain up to one billion records and each record may contain up to 255 fields, limited by a maximum record size of 4,000 bytes. The AccSys package includes a separate file, ACCSYSD.EXE, which must be available at run time, as it is used by sorting and indexing functions.

There is a comprehensive selection of functions for manipulating memo files in both dBASE III+ and dBASE IV format. These let you to create, open, close, read, write and add text to memo files. You may elect to retrieve selected parts of the memo field, starting at any point in the field. Like CB4, AccSys does not support the sophisticated editing functions available in Clipper; for this purpose you must use a text handling library or develop your own routines.

This being the multi-user version of AccSys, there are naturally some file and record locking functions. Files may be opened in either general or read-only shared mode. General shared mode allows any user to read, update or append records in files. This mode is entirely compatible with dBASE; code written using AccSys for dBASE can coexist with bona fide dBASE III+ or dBASE IV applications. Read-only

Prospero: the Pascal experts

Language Definition

Prospero Pascal is a full ISO standard Pascal, with a whole range of extensions including dynamic length strings, longreals, random access file handling, bit level manipulation, type breaking, include files and separate compilation. Prospero produced the first ever microcomputer Pascal compiler to be validated as conforming to the ISO standard (for the Z80 under CP/M) in 1983, as well as the first validated Pascal compiler for the IBM PC in 1985.

The Package

As well as a compiler, you get a linker, librarian, cross-reference generator, source level symbolic debugger (except for Z80 systems) and a free technical support hot-line.

Prospero Pascal is available for the following environments.

OS/2	£320	Full support for OS/2 and DOS. Full ISO level 1, with Prospero Workbench, the power programmer's choice.
MS-DOS	£240	As above, but without OS/2 support.
PC clones	£86.91	Personal version.
DOS/GEM	£86.91	Full GEM support, integrated development environment.
DOS→CP/M	£480	MS-DOS hosted Z80 cross compiler.
CP/M-80	£320	Amstrad version available for £49.95.
CP/M-86	£320	Including concurrent DOS.
Helios	£640	T800 transputer implementation. Includes language extensions to support parallelism.
Atari ST	£86.91	Full GEM support, integrated development environment. MC68881 co-processor library available for £60.

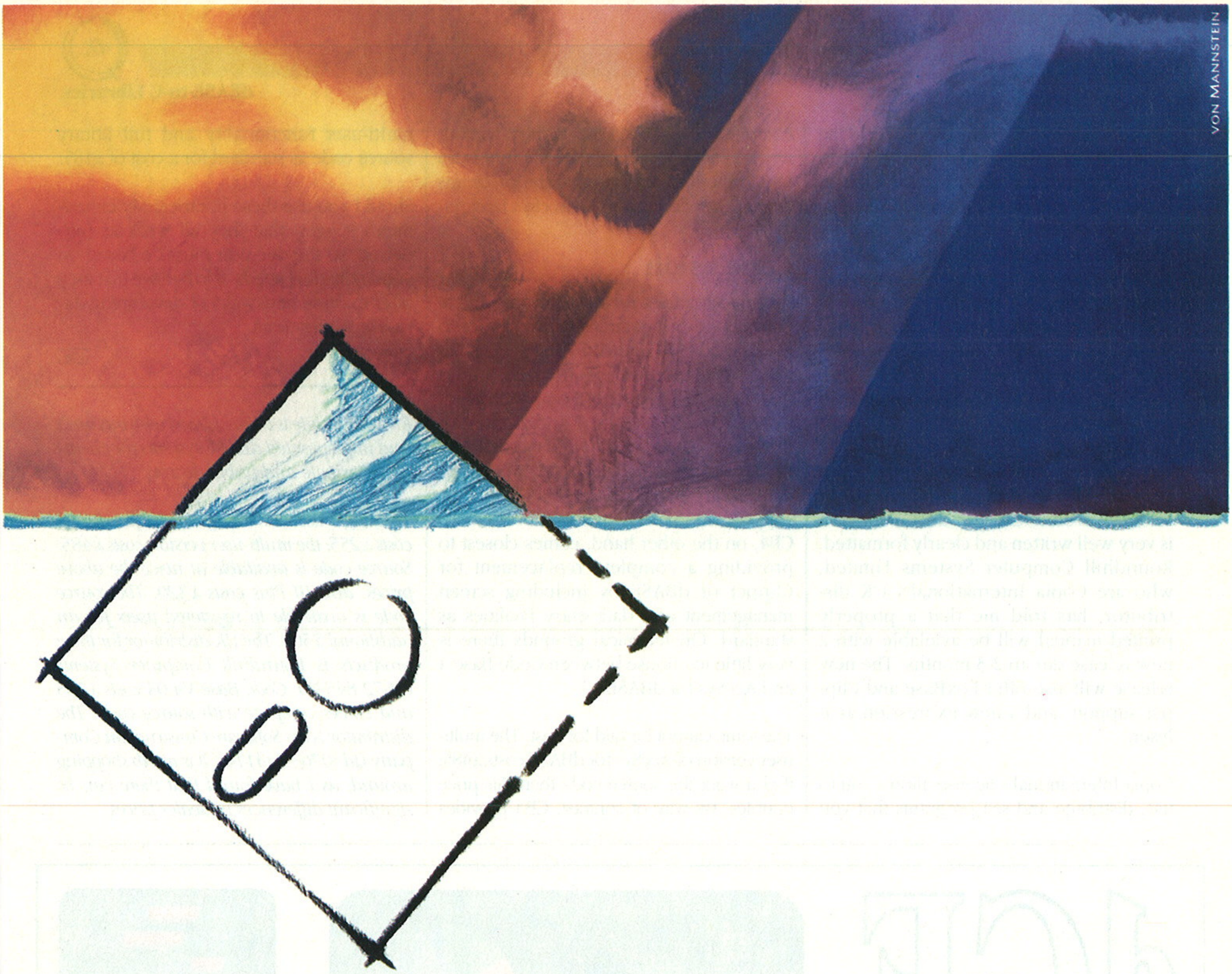
Prices include UK shipping, but exclude VAT. For information on any of these products, or any of our other development tools, contact us at the address below.

Prospero Software

LANGUAGES FOR MICROCOMPUTER PROFESSIONALS

190 CASTELNAU, LONDON SW13 9DH, ENGLAND TEL: 081-741 8531 FAX: 081-748 9344

CIRCLE NO. 134



What you know about Clipper so far is only the tip of the iceberg.

▶ And since Clipper is used by more than 250,000 developers to write PC applications, chances are that you already know many of its advantages. Like its blazing speed, its royalty-free distribution and its full support for network applications. Plus user-defined functions allowing C or other languages to be used. And an open architecture so you can channel your creativity into writing applications rather than solving shortcomings in your development system. ▶ But now Clipper 5.0 gives you even more. Like automatic memory management by a new linker which eliminates the need for overlaying. So you can run Clipper programs which are much bigger than the available memory. Like multi-dimensional arrays. Like user-defined commands providing compatibility with other data base management systems. Like on-line documentation. And all this backed by the sort of support that developers are entitled to. ▶ For a free Clipper 5.0 information pack (with details of upgrades for registered users) please clip the coupon. ▶ And once you've taken a look at what else is in Clipper 5.0, you will find it difficult to stay cool.

Nantucket UK Ltd, 8 Bridgegate Centre, Welwyn Garden City, Herts. AL 7 1 JG. Tel. 0707 373 600. Fax 0707 373 350. Also offices in Cologne, Tokyo and Los Angeles.

EXE 8/90

Name

Position

Company

Address

Phone

Nantucket®
FULL POWER

Clipper and the Nantucket logo are trademarks of Nantucket Inc.

CIRCLE NO. 135

shared mode allows many users to read the database, but only one to update it. This is more efficient than general shared mode, but not fully compatible with dBASE.

AccSys is supplied in object form, so you must specify which compiler you will use when ordering the package. The source code for the library is available at twice the cost of the object code. AccSys comes with the source code for a sample program that allow you to experiment with the library functions.

The review copy of AccSys only had a photocopied manual bound in an A4 folder. Nevertheless, the documentation is very well written and clearly formatted. Roundhill Computer Systems Limited, who are Copia International's UK distributor, has told me that a properly printed manual will be available with a new release due in 2-3 months. The new release will also offer FoxBase and Clipper support, and a new expression analyser.

Copia International's license allows you to use, distribute and sell programs that you compile using the AccSys Library without any additional licence or fees. There is no requirement to indicate that the programs were developed using the AccSys Library.

Conclusion

All three libraries reviewed in this article provide the programmer with facilities to manage dBASE files. dBC III Plus is a solid library that performs quite well, but is limited in that it is only compatible with dBASE III+. Code Base 4 and AccSys for dBASE support both dBASE III+ and dBASE IV files. AccSys for dBASE is the most up-to date library in that it is the only one to support dBASE IV's multiple index files. CB4, on the other hand, comes closest to providing a complete replacement for Clipper or dBASE by including screen management and data entry facilities as standard. On technical grounds there is very little to choose between Code Base 4 and AccSys for dBASE.

The same cannot be said for cost. The multi-user version of AccSys for dBASE costs £485; if you want the source code then this price doubles. By way of contrast, CB4 provides multi-user functionality and full library source code as standard for a cost of £185.

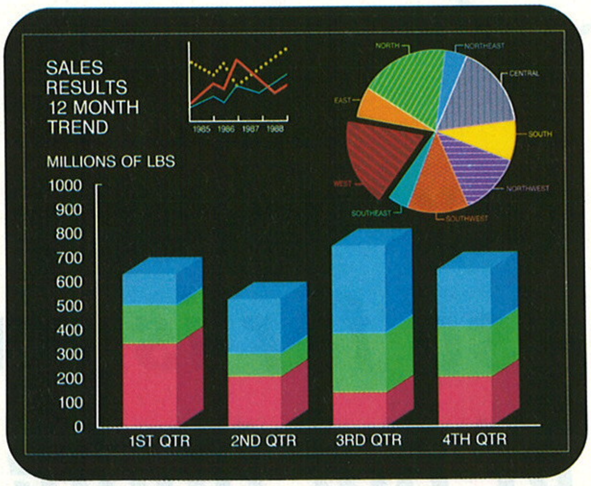
Should you use these libraries? My experiments suggest that they all work as purported, so the decision must be based on your individual needs. I do believe that they offer exciting new ways of developing database applications.

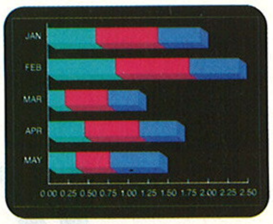
EXE

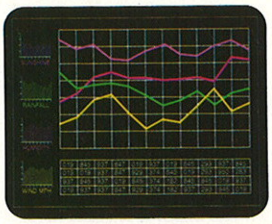
Russell Craigie is a consultant who has built and implemented dBASE systems for clients in the private and public sectors.

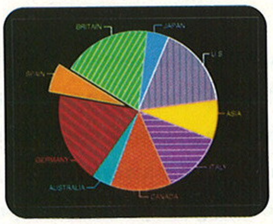
The single user version of AccSys for dBASE costs £255; the multi-user version costs £485. Source code is available at twice the above prices. dBC III Plus costs £320. The source code is available to registered users for an additional $500. The UK distributor for these products is Roundhill Computer Systems (0672 84535). Code Base V4.03 costs £185 and comes complete with source code. The distributor is the Software Construction Company Ltd (0763 244114). It is worth shopping around, as I have found that there can be significant differences in dealer prices.

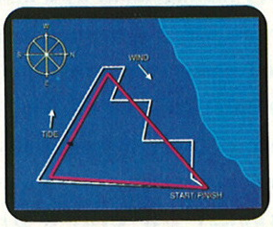
dGE
VERSION 4
GRAPHICS FOR dBASE











dGE extends the database language of your choice to give full colour graphs and charts from **within** your programs.

dGE version 3 was voted the "best graphics tool for the database manager" for the second year running by readers of the influential U.S. magazine DataBased Advisor.

Now Version 4 is here with even more features:

- SuperText with variable sizes and face styles
- SuperIcons loaded from file
- SuperText and SuperIcon editor that can import PCX images
- Built-in Vector Text functions
- Gantt charts
- XOR capability added to many functions
- New interfaces to Clipper 5, FORCE and Vulcan
- Mouse "hot" regions
- A Norton Guide to functions

dGX, the Graphics Design Centre, has been completely revised, adding:

- Sophisticated database query and summary
- Dynamic scaling
- Import of PCX images
- Statistics functions
- Gantt charts, hi-lo and 3-D bars

Version 4 comes complete with all the interfaces to dBASE and its dialects, the 'C' language, R:BASE compiler and Vulcan.

dGE supports CGA, EGA, VGA and Hercules modes and will output images to matrix printers, lasers and plotter.

Call for information.

Price £245 + VAT direct from Bits Per Second Ltd or from your authorised dealer.

dGE IS A TRADEMARK OF BITS PER SECOND LTD. OTHER BRAND AND PRODUCT NAMES ARE TRADEMARKS OF THEIR RESPECTIVE HOLDERS.

BITS PER SECOND

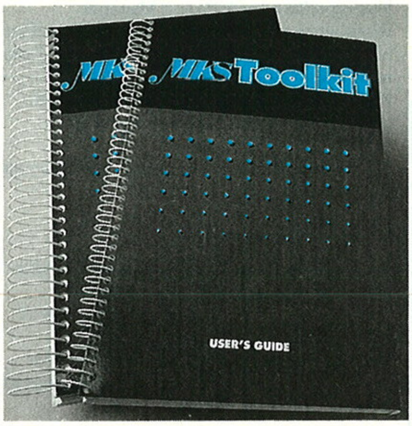
BITS PER SECOND LIMITED
14 REGENT HILL BRIGHTON BN1 3ED
TELEPHONE: 0273 727119
FAX: 0273 731925

CIRCLE NO. 136

"The MKS Toolkit is an amazingly faithful replication of a System V UNIX™ environment."

– UNIX Review

Essential UNIX Tools for any PC user



"The entire MKS documentation package proved excellent in every respect"

Daniel McAuliffe, IEEE Computer, Jan. 1990

We can tell you that MKS Toolkit offers both experts and novices the purest form of UNIX utilities in a DOS or OS/2 environment. Fortunately, we don't have to. Software reviewers, universities and major corporations all over the world are discovering MKS Software and how it can help their programmers and software developers make the most of their creative talents.

With our proprietary code, MKS Toolkit offers you more than 150 UNIX System V. Release 4 compatible tools for DOS or OS/2. With MKS Toolkit, your computer or clone becomes a comfortable environment for shells, string matching, editing, file manipulation, and more.

In this one package you will receive a complete:

- MKS Vi full – screen editor
- MKS Korn Shell – user programmable interface
- MKS AWK – programming and prototyping language
- profiler – to analyse the execution time of code
- egrep/fgrep – fast pattern matching
- archivers, sort/merge, encryption, data compression & much more

Over 150 commands that help you get the most productivity out of your PC.

Easy Installation

With MKS's menu-driven installation you can be up and running in less than 10 minutes, no matter what your level of computer expertise.

"MKS software is absolutely the best in its class. Don't mix environments without it."

Grover Righter, Novell

Interconnectivity & Lan Licences

MKS Toolkit interacts well on most standard PC and PS/2 networks. Combined with Novell Netware™, the most popular LAN for PC's, MKS Toolkit creates a UNIX time sharing system in DOS or OS/2 organizations. UNIX workgroups can now hook up all of their PCs using PC-NFS™ and MKS Toolkit, enabling you to use a PC as a UNIX workstation and off-load your mini or main machine.

Spanning Novice to Expert

This versatile software reflects it's users' needs. UNIX experts will have all the familiar power they demand while novices will find our documentation filled with excellent tutorial material.

Organizations such as AT&T, Hewlett-Packard, ITT, and NCR – all heavily committed to the UNIX system – use MKS Toolkit to create a standard operating environment. Universities from King's College to the University of Berlin use MKS Toolkit to enrich personal research computing environments and double the potential of their PC teaching labs.

"With this package, you can become familiar with the UNIX environment on your microcomputer, with DOS only a keystroke away."

Byte Magazine, May 1989

Cost-Effective Learning Tool

If your organization is committed to moving into the UNIX environment, then MKS Toolkit is the perfect learning path. DOS or OS/2 users retain the familiar world of their PC keyboard and programs yet can move to a desktop UNIX environment at a keystroke. Exposure to new commands and functionality now becomes an integral part of the novice's working day.

MKS Programming Platform

MKS Toolkit can be purchased as an individual product or you can buy MKS Programming Platform which contains MKS Toolkit and three other MKS products. MKS Platform helps smooth out the details of programming and software development by adding time-saving utilities such as: MKS RCS (Revision Control system), MKS Make (an automated program builder), MKS LEX & YACC (compiler learning and construction tools).

In all, you simply cannot find a more complete set of commands and utilities to get from DOS or OS/2 to UNIX. Once you have used MKS Toolkit, programming will never be the same again.

US List Price

MKS Toolkit (DOS & OS/2)	$399.00
MKS Toolkit (DOS/2 only)	$349.00
MKS Toolkit (DOS only)	$249.00

Please contact MKS dealers for UK domestic prices.

The Software Construction Co. Ltd. (0763) 244114; fax (0763) 244025
Grey Matter (0364) 53499
Systems Science (071) 833-1022; fax (071) 837-6411

MKS

35 King Street North, Waterloo, Ontario, Canada N2J 2W9

Put The Power Of UNIX In Your PC.

MKS, MKS Make, MKS Toolkit Korn Shell, and MS RCS are trademarks of Mortice Kern Systems Inc. UNIX is a trademark of AT&T.
PC-NFS is a trademark of Sun Microsystems.

CIRCLE NO. 137

BEST BUY

"So simple to use you wonder why other packages make such a meal of resource allocation."

PC MAGAZINE

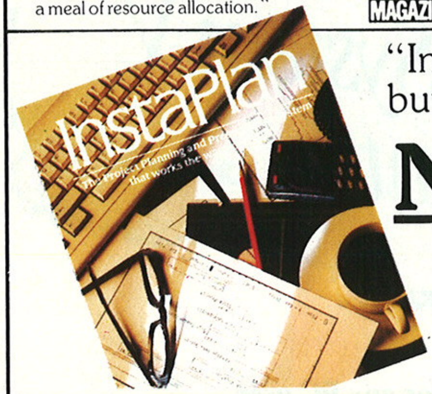
"InstaPlan offers the features of a £2,000 package with none of the trade offs of its low cost competitors."

PERSONAL COMPUTER EDITOR'S CHOICE

"We would recommend most warmly... InstaPlan as the best full featured project management package."

Practical Computing

(compared with PMW, Super Project and Harvard).



"InstaPlan is excellent, much better value than PMW, but keep improving it" Mr D Symington, CS Consultants

NEW InstaPlan 5000

InstaPlan 5000 is the latest addition to the InstaPlan family of project management software, which already has 25,000 users, from large multinationals to one man businesses, and is fully compatible with the rest of the InstaPlan family.

Easier Departmental Management

InstaPlan 5000 can gather individual project plans and consolidate them for departmental review, all with a single command on a stand alone machine or a PC network.

By bringing together the department's schedules and resource requirements in one model, you can recognise and resolve staffing conflicts and project priorities long before they become crises.

Most project management systems require all projects to be updated to the same date before merging. This is often impossible. InstaPlan 5000 allows projects to be combined freely.

Planning and Control for Larger Projects

InstaPlan 5000 will run on a standard 640K PC, Just like the rest of the InstaPlan family. But InstaPlan 5000 also supports EMS (Expanded Memory). This means that you can handle in excess of 10,000 Activities. Within these limits you are only constrained by the amount of EMS memory you have available. This is many times the capacity of some of InstaPlan's more expensive competitors.

Better Resource Management

You can now level the load on a group of resources simultaneously, using he Adjust Loading command. The Locate Overload facility will scan all future periods for a resource, identifying any overloads.

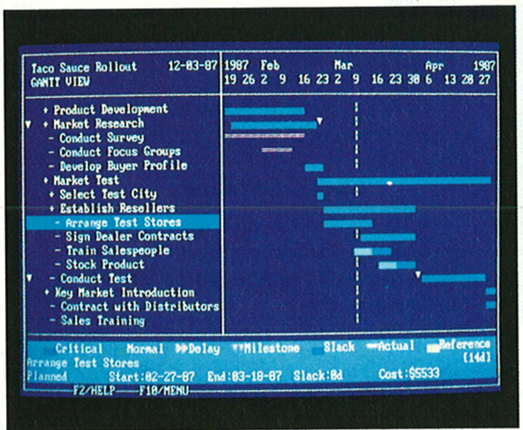
We believe InstaPlan 5000 will provide the best solution for professional project managers and busy line managers who need project management to enhance their own and their department's performance.

See what InstaPlan users say:-

Easy to learn and Use

...it's design is just right. I set up a workload plan without even using the manual or any previous planning experience.

David Allen — *Computer Services Manager*
Vallectric Ltd — *Electrical Contractors*



With Quality Documentation

...a very good package, with excellent documentation.

Mr J Clarke — *Finance Director*
Ceraf Ltd — *Building Contractors*

...And Support too

...Good telephone support
Dr Robin Wilson
Harrogate Borough Council

Providing the Right Reports

...provides me with succinct and easy to understand documentation for Board presentations.

Dr R Gailey — *Director of Research*
Tootal Thread Ltd

For Line Managers

Even though our busy line managers are only occasional users, InstaPlan gives them the tools to plan quickly and effectively.

Brian Alldridge — *MIS Manager*
Prestwich Holdings plc

and for Project Managers

I am surprised to find myself learning from this product – an excellent investment.

Mr P M Cleary — *Project Manager*
Maritime Dynamics Ltd

compared with the competition

Having suffered with Project Manager Workbench and SuperProject Expert, InstaPlan has provided us with the tools to produce all our reports and regular project updates easily. Management are impressed by the ease with which they can read and understand the reports.

Ms Petra Meyer — *Product Manager*
Travicom Ltd

In summary

InstaPlan contains all the features and facilities required for project tasking, costing and tracking. An excellent, good value package, it forms the basis of successful project management within a business environment.

Andy Coster — QA Manager,
Logsys

Money Back Guarantee

We are confident InstaPlan 5000 is the easiest project management system to learn and use, offering a significant price/performance advantage over all other systems. Buy InstaPlan 5000 now, if you return it within 30 days for whatever reason, we guarantee to refund your money by return.

DEEPAK SAREEN ASSOCIATES

Freepost, 28 Preston Rd, Wembley HA9 8BR
Phone 01 908 2423 Fax 01 904 6668

Instaplan 5000 £495 plus £5 p&p and VAT

☐ **Please send me InstaPlan 5000.**

☐ **I enclose a cheque for £575.00**

☐ Please bill my Visa/Access Card

☐ **send me more details of InstaPlan 5000 and the InstaPlan family, from only £99.**

Company ....

Name..

Signature ....

Phone no. ....

Please enclose a compliments slip showing your required delivery address and credit card details if appropriate.

EXE 6/90

Configuration: IBM XT/AT PS/2 or compatible running under PC/MS-DOS 2.1 or later with at least 500K of DOS memory and 3M of hard disk available to InstaPlan. Plan storage requires further DOS memory or access to EMS memory. EMS memory (LIM3.2+) supported to a maximum of 8Meg. A 5000 activity plan will require at least 2 Meg of expanded memory. InstaPlan 5000 will therefore run across networks using network shells like Novell where there is sufficient DOS memory available to InstaPlan. Printers: IBM ProPrinter & Graphics Printer, Epson FX/LX/LQ, Toshiba 351, HP LaserJet PLUS/Series II/Deskjet. Input via keyboard, ASCII, ThinkTank or READY! Output to ASCII, Lotus or dBase. Charts are stored as bit mapped graphics and can be printed but not viewed on screen. ALL MARKS ACKNOWLEDGED. DEALER ENQUIRIES WELCOME. QUANTITY DISCOUNTS AND SITE LICENCES AVAILABLE TRAINING AND CONSULTANCY AVAILABLE IF REQUIRED.

CIRCLE NO. 138

Driving Clipper to abstraction

One of the big drawbacks of the Clipper language is the curious set of scoping rules. Mike Hart suggests a small suite of C programs, with just a hint of OOP, as a remedy.

Data abstraction (or 'encapsulation') is one of the most important components of the object-oriented approach to programming. Although the Clipper language supports some of the features of the earlier structured programming revolution, it lacks any facilities to support OOP. Even worse, the sloppy scoping rules positively hinder it.

Objects

To see why this is, we must once more go over the definition of an object. An object has a state, usually implemented as a data structure of some sort, and a number of associated functions ('methods') which can manipulate it. The state of the object must be maintained between calls to these functions.

A familiar example is the common or garden disk file. In C, for instance, there is a fixed set of I/O library functions which offer a consistent interface to the programmer. Implementation details vary between machines and operating systems. The programmer, however, only 'sees' the file through this set of functions. This arrangement hides the structures used to implement the file, which can be changed as required without affecting the client programs. It also 'modularises' the file system: if you do need to make changes, you know where to go to make them and it does not affect other parts of the program.

Data structures are protected, since they cannot be altered, except by the designated functions - hence the integrity of the object is guaranteed. Finally, the 'plug in' separateness offers the possibility of reusable software. General purpose objects can be built and used in many different situations.

This hiding and protection of program structures is called Data Abstraction and the objects so created Abstract Data Types (ADTs). I will refer to ADTs as 'objects', although strictly speaking they are not, because they do not have the property of inheritance, demanded by most definitions of OOP. The creation of objects with the property of inheritance is beyond the scope of this article. Those who complain of abuse of the term 'object-oriented' in this context should note my defence: the code presented here makes it easier (if not easy) to use an object-oriented design method; after all, the design technique should work whatever the qualities of the target language.

Design for objects

In order to build programs with objects, we need a new design approach. In functional decomposition, the design proceeds by 'divide and conquer', dividing the problem into ever more manageable functions. Object-oriented design, on the other hand, requires the identification of real world entities (OOP's simulation roots showing here) plus the actions associated with them. The objects so discovered may themselves be further subdivided. Often the low-level objects are general purpose services for client programs: items such as stacks and queues.

Programs designed in this way exhibit the advantages described above. The modules are generally less coupled and are easier to change. Also, it is usually the case that the more closely your program structures model their real world counterparts, the more likely it is that the impact of change can be minimised, or at least be analogous in size to the real world changes being modelled.

The Clipper Extend system

Clipper includes an interface to C, allowing C functions to be called from your Clipper programs. Since Clipper was written in Microsoft C V5.1, you will save yourself much grief by so using this version of C when linking to Clipper.

The C interface is specified in two header files, NANDEF.H and EXTEND.H, which must be `#included` in your C source files. These files contain macros, constants and function prototypes which allow you to pass parameters to and from Clipper, which uses non-standard conventions for passing and returning arguments. All the C functions called from Clipper must be declared as type `CLIPPER` and have no arguments.

To determine the number of parameters passed from Clipper, the macro `PCOUNT` is used. There are a set of macros to test the type of a parameter, for example `ISNUM()` and `ISCHAR()`. These macros take the parameter number, as read left to right in the Clipper source, as their argument. The parameter values are obtained by the `_par...()` family of functions - there is a different one for each supported type. In my code, integer parameters are captured with `_parni()`, longs with `_parnl()` and strings (characters) with `_parc()`. Returns to Clipper are achieved via the similar `_ret...()` functions: `_ret()` returns nothing, `_retni()` returns an `int`, `_retnl()` a `long` and `_retc()` a pointer to `char`.

This interface is a bit limited. For example, there is no mechanism for integer parameters to be passed by reference. This is why the `st_put/st_get` functions only pass one value at a time.

To compile the C program, the large memory model must be used and the library header information left out. This gives a compile command that looks like this:-

```
cl /AL /c /Ox /Zl statics.c
```

The resulting OBJ file can be linked into Clipper programs in the normal way, except that the /NOE flag must be used.

Clipper objects

To implement an object, we must be able to maintain the state of its structures (its instance data) between calls to the methods. There are a number of ways of doing this using the Clipper language, but they all present serious disadvantages.

The most obvious approach is to declare the data as `PUBLIC`. This is a very dangerous way to proceed, especially if the public variables are embedded in the code. One of the qualities we are striving to emulate is the protection of variables from access by unauthorised code. Declaring `PUBLIC` variables has exactly the opposite effect.

An alternative is to declare the data as `PRIVATE` to the environment which calls the 'method' functions. As you will recall, with Clipper this means that code at the declaration level and all functions below can see the data, unless further `PRIVATE` variables, with the same names, are created at some lower level. Data declared in this way ties two logically separate entities together (the application program's code and the object's code).

Worse, if you need to use more than one object in the same scope, you will have to pass all the data structures as parameters. These quickly become confusing and, because Clipper does not verify parameters at compile time, often result in run-time errors.

Finally, if you make changes which affect the data structures, you must search through the code for each occurrence of them. This is exactly the situation that objects are designed to avoid.

One last possibility is to create one carefully named `PUBLIC` variable and use it to hold all the structure values which need to be maintained. These values can be loaded into local variables when one of the allowable functions is invoked and saved just before return. The `PUBLIC` variable can be given a 'sure to be unique' name created at run-time.

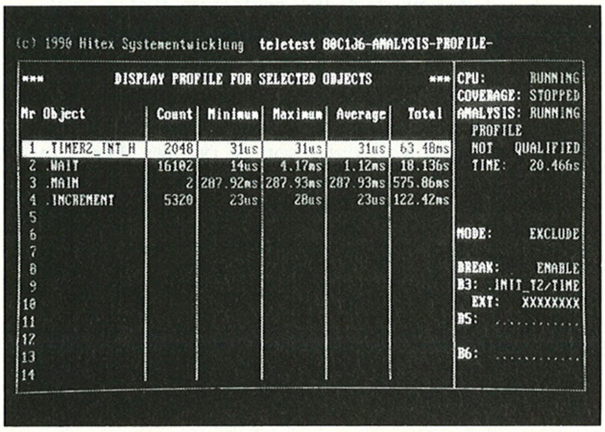
This method mostly avoids the problems of using public variables. It also maintains the state of the object within the object's scope, so it shows promise. The major problem with this approach is its sluggishness - testing shows it to be hopelessly slow.

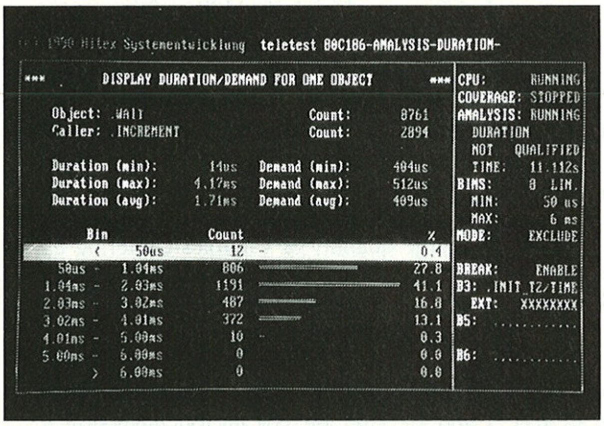
```
/* STATICS.C */

#include <malloc.h>
#include <string.h>
#include "nandef.h"
#include "extend.h"

#define NAME_MAX 12 /* names up to 11 chars */
#define NUMBER(n) (n == -1)
#define STRING(n) (n >=0)
#define NUMTYPE -1
#define EMPTY -2
#define INVALID -1

/* structure holds variables of two types supported */
struct var_rec{
 int length;  /* length -1 its a number */
 union {     /* length >=0 its a string */
  long number;
  char *string;
 } var;
};

CLIPPER st_create(void);
CLIPPER st_put(void);
CLIPPER st_get(void);
CLIPPER st_dispose(void);
char *install(long);
void name_put(char *,char *);
int search_for(int,char *,char *);
int valid_addr(struct var_rec *);
char *strdup2(char *);

/* create/initialise a new set of static variables */
CLIPPER st_create()
{
 int i,char_ctr;
 long ctr; 7* count of number of static variables */
 struct var_rec *ip,*ip2;
 char *char_ptr;
 ctr = (ISNUM(1)) ? _parnl(1):PCOUNT; /* num = array mode */
 ip = (struct var_rec *)
       malloc((ctr+1)*sizeof(struct var_rec));
 for(i=0,ip2 = ip;i<=ctr;i++,ip2++) ip2->length = EMPTY;
 ip->length = ctr; /* 1st record holds count of records */
ip->var.string = install(ctr);
 _retnl((long) ip);
}
/* store the names of the static variables */
char *install(long size)
{
 int i;
 char *names,*tp;
 if(ISNUM(1)) return NULL;
 names = (char *) malloc(size*NAME_MAX);
 for(i = 0,tp = names;i <PCOUNT;i++,tp+=NAME_MAX)
  name_put(tp,_parc(i+1));
 return names;
}

/* put the next name in the allocated space */
void name_put(char *dest,char *source)
{
 int i;
 for(i = 0;*source && i < NAME_MAX-1;i++)
  *dest++ = *source++;
 *dest = '\0';
}

/* put a new value in a static variable */
CLIPPER st_put()
{
 int i;
 struct var_rec *base;
 base = (struct var_rec *) _parnl(1);
 if((i = valid_addr(base)) != INVALID){
   base+=i;
  /* if you are adding a string to a string and it fits */
  if(ISCHAR(3) && STRING(base->length)
        && _parclen(3) <= base->length)
    strcpy(base->var.string,_parc(3));
  else if(STRING(base->length))
 free(base->var.string);
  if(ISCHAR(3)){
 base->length = _parclen(3);
 base->var.string = strdup2(_parc(3));
  }
  else if(ISNUM(3)){
 base->var.number = _parnl(3);
 base->length = NUMTYPE;
  }
 }
 else  _ret();
}

/* get a value from a static variable */
CLIPPER st_get()
{
 int i;
 struct var_rec *base;
 base = (struct var_rec *) _parnl(1);
 if ((i = valid_addr(base)) != INVALID){
   base+=i;
  if (STRING(base->length))
 _retc(base->var.string);
  else  /* must be number */
 _retnl(base->var.number);
 }
 else _ret();
}

/* free up the memory used by the statics */
CLIPPER st_dispose()
{
 int i;
 char c;
 struct var_rec *base,*tp;
 base = (struct var_rec *) _parnl(1);
 if(base->var.string != NULL) free(base->var.string);
 for(i=1,tp=base;i <=base->length;i++)
  if(STRING((++tp)->length)) free(tp->var.string);
 free(base);
}

/* find the offset for static variable with value 'name' */
int search_for(int size,char *nametab,char *name)
{
 int i;
 for(i=0; i < size && strcmp(nametab,name)!=0;
             nametab+=NAME_MAX,i++);
 return (i < size) ? ++i:INVALID;
}
/* check that requested offset is valid */
int valid_addr(struct var_rec *p)
{
 if(ISNUM(2) && _parni(2) >=0 && _parni(2) < p->length)
   return _parni(2)+1;

 if((p->var.string != NULL) && ISCHAR(2))
   return search_for(p->length,p->var.string,_parc(2));

 return INVALID;
}

/* Clipper does not recognise strdup so this is a replacement */
char *strdup2(char *source)
{
 char *c,*c2;
 c = (char *) malloc(strlen(source));
 strcpy(c,source);
 return c;
}
```

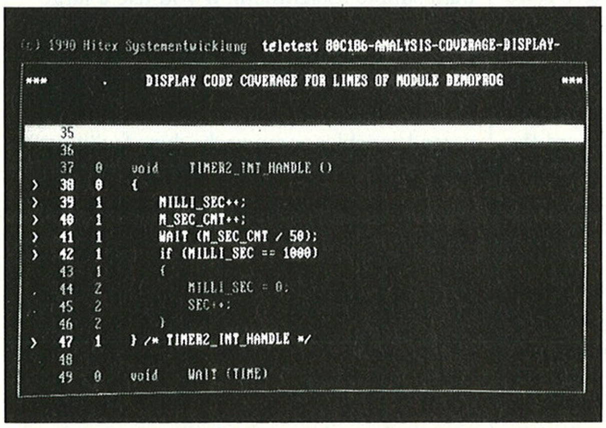
Figure 1 - The STATICS.C file to implement Clipper objects.

HiPAT

Performance Analysis Tool For 8086 Family Systems







If you can answer YES to all the following questions, then you don't need *HiPAT*:

1. My program is coded so well that it cannot possibly run any faster.
2. There is no "dead" code - every byte is used.
3. I am absolutely certain that all variables are initialised before use.
4. I know exactly which procedures take the most time.
5. When interrupted, the latency time of my program is under xx ms.

But if, as in the real world, the runtime performance of your compiled 80x86 system is not quite up to scratch, *HiPAT* is the solution.

HiPAT is a hardware CASE tool which allows a complete real-time analysis of Microsoft, Borland and Intel compiled programs, running on the 80x86 processors.

HiPAT will reveal program linkage, functions' runtime distributions, redundant code and variables, plus it will implement single and multiple object analysis thus allowing program optimisation to be precisely targeted and data usage errors eliminated.

HiPAT is PC-hosted and supports 8086/186/286/V40/V50.

Call us now for a free demonstration disk!

Hitex (UK) Ltd., Warwick University Science Park, Coventry CV4 7EZ Tel. 0203 692066

hitex

Experience in High Technology

CIRCLE NO. 139

Even the new features of Clipper 5 will not save the day. Although the new version of Clipper will have 'static' variables (which retain their value between successive calls to functions), they will be *local* statics. This still makes it difficult to implement an ADT, which usually requires multiple functions to have access to the 'static'.

Fortunately, Clipper has a bridge into C. Using the extend functions (see separate box) it is possible to create variables which maintain their values and are not prohibitively slow in use.

The functions

The idea is to borrow C's dynamic allocation to create variables which can then be accessed from Clipper. As these are dynamically allocated, they persist until freed by the programmer and can, therefore, maintain a value across calls to functions.

Time to present my library. The code for the functions is shown in Figure 1. To explain function usage, I will use the common notation of angle brackets for non terminals, the bar for OR and ellipses for repetition. The first function has the following syntax:-

```
st_create(<string exp> ...
     | <numeric exp>)
```

The function allows you to declare how many variables you want. You can either 'name' the variables by listing them as string parameters to st_create:

```
* STACKS.PRG - generic stack program
* demonstrates the use of STATICS.C

external st_create, st_put, st_get,st_dis-
pose && from statics

function stk_create
* create a new stack
parameters maxsize
private stack
  stack = st_create(maxsize+2) && extra
place for head + size
  * some check that stack <> to 0 (NULL)
here
 st_put(stack,0,2) && use pos 0 to remember
head initialise it
 st_put(stack,1,maxsize+2) && use pos 1 to
remember stack size
return stack

function push
* push an item onto the stack
parameters stack,item
private head
  head = st_get(stack,0)
  if head > st_get(stack,1)
    RETURN .F. && or possibly crash the
program
  end
  st_put(stack,head,item)
  st_put(stack,0,head+1)
RETURN .T.

function pop
* pop an item from the stack
parameters stack
private head
  head = st_get(stack,0)-1
  if head < 2  && stack proper starts at
position 2
   RETURN "" && or crash the program
  end
  st_put(stack,0,head)
RETURN st_get(stack,head)
```

Figure 2 - The STACKS program, an ADT based on STATICS.

```
object =
st_create("var1","var2","var3")
```

or you can use 'array mode':

```
object = st_create(20)
```

which creates 20 anonymous variables. If you use array mode you can only refer to variables by their offset. The first offset is '0', the next '1' etc. If you create named variables you can refer to them either by name or offset (their ordinal position in the declaration list).

Variables are stored by using

```
st_put(<object>,
     <offset | name>,<value>)
```

`<Name>` is a string expression, `<offset>` is a numeric expression, `<object>` is the value returned by `st_create`, and `<value>` is what you want to store. This may be a string expression or numeric expression. As with any normal Clipper variable, the type of value is not fixed so, even if you have used a particular variable as a string, you can go on to use it to store a number.

Values are retrieved using

```
<var> = st_get(<object>,
          <name | offset>)
```

parameters/returns similar to `st_put`. The functions perform a range check on offsets, and will return False if a non-matching name is passed in. If you `st_get` from a variable which has not been `st_put`, the returned value is rubbish. Finally,

```
st_dispose(<object>)
```

is provided to return the objects' memory to the heap when they are no longer required.

The example in Figure 2 shows how the set of functions can be used to create an ADT, in this case a stack. Notice that there is no need to declare any structures outside of the ADT. To create a stack, call `stk_create(digit)`, where `digit` allows you to specify the maximum size of the stack. (It would have been possible to store the statics as a linked list, and so create a variable length stack. However, I have shown a simpler fixed-size stack for the sake of clarity and brevity.)

The `stk_create` function returns a pointer to the new stack object. This pointer must be used to access individual items on the stack - it serves as a means of distinguishing a particular instantiation of the stack. As a result, it is possible to create as many stack objects as required. The `push` and `pop` functions work as expected.

Criteria met?

The library of routines discussed above do provide the required modularity. If we need to alter the internal workings of the stack presented in the example, there would be no need to look outside the STACKS file. The structure of the ADT is hidden from clients. The stack is an example of a reusable library module, and, most important of all, it allows the creation of something near enough to an object to ease the use of object-oriented design.

But we lack the protection that a 'real', language-supported, ADT would offer. There is no concept of 'type' in Clipper, so it is not possible to distinguish the object pointer from an ordinary integer, far less from a different sort of pointer. You can cheerfully perform arithmetic on the pointer without a squeak from Clipper - until you try to use it as a pointer again. The only protection that we have is the scope of the pointer, which is restricted to the client program.

Things would be much better if we could protect the returned pointer from change, and also plug the other main gap in the defences, which is the ability of the user to manipulate objects by calling `st_put` and `st_get` directly, instead of via 'allowable' functions such as `push` and `pop`. For the time being, we must depend upon good programming style and a sensible attitude from the user. We have, I believe, a better level of protection than achieved by using a large quantity of variables cluttering up the client program, waiting to be assigned values inadvertently, or passed in the wrong place in the parameter list.

There is, of course, a speed penalty. `st_get` and `st_put` are typically four times slower than simple assignment. If you use a large number (20 or more) of named variables, and access them by name, this rises to about four and a half times. It is possible to write a streamlined version of STATICS.C (no named variables or range checking facilities) but this produces timings only about three and a half times slower so probably does not justify the extra risk and inconvenience

Conclusion

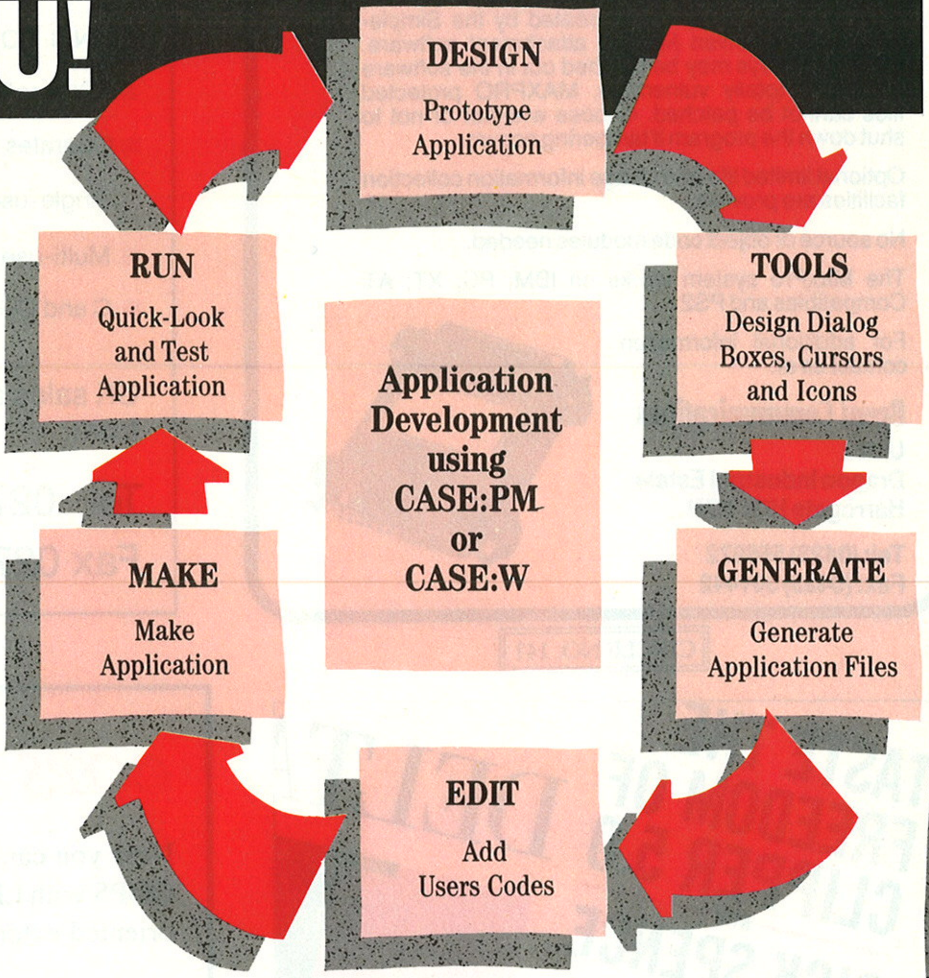
The STATICS functions do not have the total security that we would like from an abstract data type, but they offer higher security than other methods. Program modularisation is also improved. There is a trade-off of speed against design benefits, but the overhead is not prohibitive.

EXE

Mike Hart has a degree in computer science from the Polytechnic of Central London and is now an independent consultant.

CASE:PM or CASE:W

THE TOOLS THAT WRITE PM OR WINDOWS CODE FOR YOU!



Writing applications for Windows or OS/2 Presentation Manager graphical user interface just became a lot easier, because now there's CASE:PM (for Presentation Manager) and CASE:W (for Windows).

CASE: PM™

Both are application prototypers and code generators that write complete C skeleton programs for you based on a description of the 'user interface'.

APPLICATION PROTOTYPING

Just by 'pointing and clicking' at various parts of a window you can now create a Windows or PM user interface – in minutes!

APPLICATION GENERATION

Once you have the 'look and feel' of the application, CASE:PM or CASE:W produces C source code, resource scripts, definition files and make files… *automatically*.

Run *MAKE* and the files are compiled and linked producing a standalone executable with no run-time environments or royalties to pay.

CASE: W™

APPLICATION MAINTENANCE

Once you have the skeleton you add the application specific code. The generated source code is fully documented, with instruction as to where to put the application specific lines, leaving you free to concentrate on the application itself.

Change the user interface design and CASE:PM or CASE:W will generate the source files with the new user interface – but retaining the code you've added.

CASE:PM or CASE:W will reduce your learning curve and increase your productivity. For information and technical specifications call the Products Group on 0285 655888 FAX (0285 650537), or write to us.

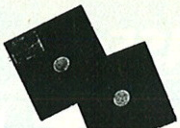
QA

Cecily Hill Castle
Cirencester, Gloucester GL7 2EF
0285 655888

CIRCLE NO. 140

BRENT
Communication and Telemetry Consultants

"MaxPro – the ultimate in hardware for software protection"



The MaxPro Data key fitted to the Parallel Port is totally transparent to the end user.

Software customers make as many backups as are required but of course only the machine using the individually coded MaxPro key will run the program.

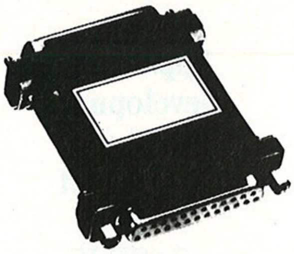
BEWARE THE PIRATE'S PATCH
High security encryption is created by the Simple-to-Use Menu-driven MaxPro attachment software. Other data keys may be patched out in the software rending it totally vulnerable. MAXPRO protected files cannot be patched. Choose whether or not to shut down the program if tampering occurs.

Optional limited life and Usage information collection facilities are provided.

No source or object code modules needed.

The MaxPro system works on IBM; PC; XT; AT Compatibles and PS2.

For additional information contact us at:



Brent Communications
Unit 2
Dragon Industrial Estate
Harrogate HG1 5DN

Tel: (0423) 566972
Fax: (0423) 501442

CIRCLE NO. 141

SQL FROM xdb

The high performance extended relational database management system that features:

- ANSI, SQL/DS and 100% DB2 compatibility
- 4GL application tools
- Operates within the 640K DOS boundary
- Single-user version for DOS, OS/2
- Multi-user network versions for DOS and OS/2
- C and COBOL application interfaces

UK sales and support from:

CRM
COMPUTER RESOURCE MANAGEMENT LIMITED

Tel 0272 570516
Fax 0272 575825

CIRCLE NO. 142

TASTE THE FREEDOM OF CLIPPER 5.0

WITH

RICK SPENCE

Let Rick Spence, Clipper's Co-Developer, stretch your imagination with his unique pre-release overview of the eagerly awaited new Clipper 5.0 in one day seminars at Delta's London training centre on 4th, 5th and 6th September. Numbers are strictly limited to 15 each day so book now by phoning us on 071-831 6994.

THE NEW CLIPPER 5.0 TRAINING PROGRAMME

Delta Dimensions follow up with the launch of their new, comprehensive range of Clipper 5.0 training courses immediately the product is released.

Nantucket AUTHORISED TRAINING CENTRE

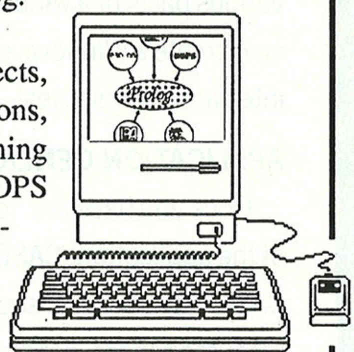
DELTA DIMENSIONS

DELTA DIMENSIONS LIMITED
20-21 TOOKS COURT, LONDON EC4A 1LB
TELEPHONE 071-831 6994

CIRCLE NO. 143

OOPS + AI -> Prolog++

Now you can combine the best features of AI and OOPS with LPA's new Prolog++, the all new object-oriented extension to Prolog.



Prolog++ integrates objects, methods, messages, functions, attributes - in short, everything you would expect in an OOPS environment - with the declarative power of Prolog.

All of the standard features of LPA Prolog, such as source level debugging, incremental compilation, sophisticated graphics and windows, and so on, are inherited by Prolog++.

So why not be objective, and give us a call?

LPA
Logic Programming Associates Ltd
Studio 4, RVPB, Trinity Road, London,
SW18 3SX, England
Phone: 081 871 2016
Fax: 081 874 0449

CIRCLE NO. 144

A linker called Blinker

Blink Inc's Blinker linker has already impressed the Clipper world - with its name! But does it herald a new, carefree era for RAM-caged developers? James Ormrod investigates.

Linking has long been a source of discontent among Clipper developers. The early announcement of Clipper 5.0, to be bundled with Pocketsoft's .RTLink, has served to heighten awareness and introduce the concept of dynamic overlays to developers. With the demand so skilfully created, all that was needed was a supply. Blinker got to market first and has thus been able to take advantage of this demand. But is it any good?

Static vs dynamic

Remember how it was with PLINK86? Your application grew and grew, it ATE memory faster than Donald Trump's overdraft eats emergency supplementary loans. You avoided the issue as long as possible, but sooner or later those `MEMOEDIT()`s and `DBEDIT()`s started to crash. Of course, they didn't crash all the time, just now and again, usually in the presence of the Customer. So you resigned yourself to overlays. Static overlays, as implemented by PLINK86, force the developer to identify mutually exclusive chunks of code, compiling them into separate object modules. These modules are combined into overlay units; the smallest overlay is a single object module. Space is reserved for overlays in the .EXE program file based on the size of the largest overlay unit or maximum combination of nested overlays. When code within an overlay is called, the overlay manager loads the appropriate overlay unit into the reserved space and passes control to it. If code in the overlay references symbols within other overlay units the results can be unpredictable (=fatal). The size of the buffer reserved for loading overlays is fixed by the size of the largest overlay unit, so static overlays waste memory unless all the units are the same size.

Contrast this with Blinker's system of overlays. The dynamic overlays used by Blinker work at the procedural level, each individual procedure and function being treated as an overlay unit. Each overlay unit is treated as if it has the potential to call every other unit, the Blinker overlay manager resolving external dependencies from a fix-up table stored within it. Dynamic overlays are loaded into a resizable overlay pool within Clipper's free pool, using the minimum amount of memory. When the application is run, the overlay manager loads overlays into the overlay pool in the order in which they are called. When the overlay pool is full, a request to load another overlay causes the manager either to discard overlays on a least-used/oldest basis, or, if all current overlays are actively in use, to increase the size of the overlay pool. The technique transfers the burden of dealing with overlays from the programmer to the software, where it belongs.

Figures 2-4 show example memory maps for 'straight' (non-overlaid), static overlaid and dynamic overlaid versions of an application. I suggest that if you are unfamiliar with overlays, you take a quick squint at these to clarify what goes on inside the machine when a Clipper application is loaded.

The package

Blinker V1.11 is supplied on single disk containing BLINKER.EXE (the linker itself), BLINKER.NG (Norton Guide help - *you* supply the Norton Guide engine), the usual last minute additions to the manual READ.ME and source code for an error system and a sample profiler function. Installation is left to the user, on the assumption that a developer should be able to cope with the MS-DOS COPY command.

The manual is a slim (100 page) A5 ring binder that contains all you need to know, even if you sometimes have to hunt for it. The layout is intended to make it usable as both a tutorial and a reference manual (you are encouraged to read it straight through before you start using Blinker), although it succeeds more in the former and could do with a quick reference section. There is considerably more to Blinker than most other linkers, but still not really enough to warrant a Norton Guide, and the one supplied is not fully cross-referenced.

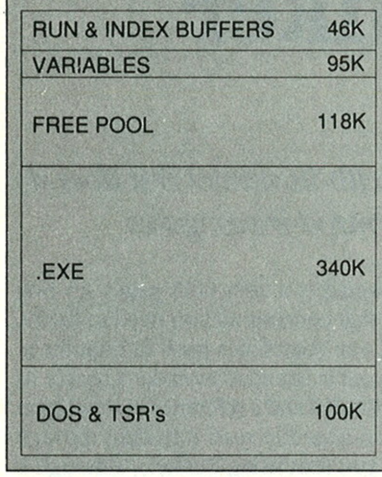
Blinker can be used to perform a 'straight' link, without creating any overlays. In practice, this is a perverse thing to do, as the resulting .EXE file will carry the 5 KB overhead of the overlay manager without using it. In order to build overlays, Blinker needs to be told which routines to place in the permanently resident .EXE root, and which should be dynamically allocated. This information is supplied in a script file, which usually has a .LNK extension.

Blinker script files are similar to PLINK86's (most Blinker command syntax is compatible with PLINK86). Blinker makes two passes through each file, processing all FILE commands on the first pass and everything else on the second. You cannot nest script files, but Blinker allows you to specify more than one on the command line, a feature that could be useful when creating different versions of an application.

```
FILE MENU                  # The root module
LIB ASM_LIB                # Cannot ALLOCATE, calls interrupts
BEGINAREA
  FILE MAINTAIN            # Overlaid Clipper code is all lumped
  FILE REPORTS             # together with no nesting required.
  FILE HOUSEKEEP
  ALLOCATE EXTEND          # Well behaved add-on libraries,
  ALLOCATE MYLIB           # including user-defined and C/ASM
  ALLOCATE C_LIB           # libraries can be overlaid.
ENDAREA
LIB CLIPPER                # Last to correctly resolve duplicates
```

Figure 1 - Typical Blinker script file.

Notes:
EXE load module includes string literals in the source code as well as a separate symbol table for each component OBJ file. You can reduce the load requirements by rationalising strings (replace `"-----"` with `REPLICATE("-",5)` etc) and compiling all source into a single OBJ.
The free pool is used for manipulating data and is the area that suffers from fragmentation. String and array data is stored here and macros extensively will manipulate them within the pool (`DBEDIT()` for example).

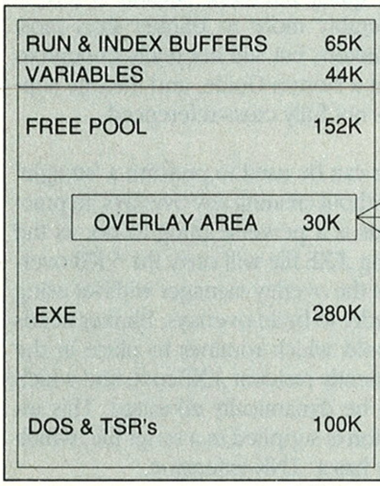


Each variable has a 22 byte entry reserved for it, consisting of its name and either its data or a pointer to its data.
The RUN and index buffer area is used in building indexes and loading a DOS COMMAND shell when the RUN (or !) command is issued.
You can claw back some memory with the SET CLIPPER environment variable, reducing the RUN area to 16KB, for example, will slow down index operations but release an extra 30KB, which will be added to the Free Pool.

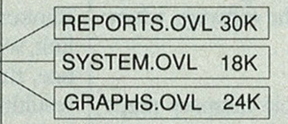
Calculations:

640KB - 100KB	DOS & TSR's (typical)
540KB - 340KB	reported EXE load size
200KB - 24KB	initial free pool size
176KB 176*20%	memory variables (35KB)
141KB - 141*33%	RUN/index buffers (46KB)
94KB + 24KB	final free pool (118KB)

Figure 2 - Clipper memory allocation, no overlays.



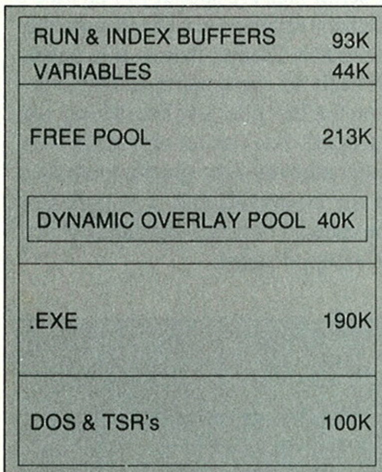
Notes:
The largest overlay dictates the size of the overlay swap area. Internal overlays are slightly more efficient. They are tacked onto the bottom of the EXE file instead of existing as separate disk OVLs.



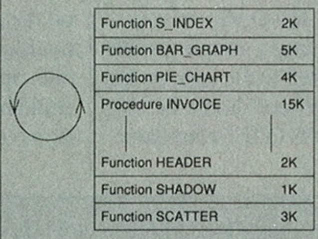
Calculations:

640KB - 100KB	DOS & TSR's (typical)
540KB - 280KB	reported EXE load size
260KB - 24KB	initial free pool size
236KB - 236*20%	memory variables (44KB max)
192KB - 192*33%	RUN/index buffers (63KB)
128KB + 24KB	final free pool (152KB)

Figure 3 - Static Overlays.



Notes:
Every function and procedure is treated as a separate overlay unit and loaded or discarded dynamically as the program executes. The overlay pool is reserved by BLINKER and can be altered. You can still use SET CLIPPER to expand the free pool; in this example we could claw back an additional 99K! (16KB for RUN/index saves 77KB while 22KB from the variables area still leaves enough for 1024 variables, which is plenty).



Calculations:

640KB - 100KB	DOS & TSR's (typical)	326KB - 44KB	memory variables (44KB max.)
540KB - 190KB	reported EXE load size	282KB - 282*33%	RUN/index buffers (93KB)
350KB - 24KB	initial free pool size	189KB + 24KB	final free pool (213KB)

Figure 4 - Dynamic Overlays.

Figure 1 shows a sample Blinker script file. The `BEGINAREA..ENDAREA` structure identifies the overlay code, note the `ALLOCATE EXTEND` command which overlays the Clipper `EXTEND` library. Root code is identified outside the overlay area, with the entry module specified first. Code that cannot be overlaid (ASM or C code that calls interrupts, for example) and un`ALLOCATE`able third party libraries come next, CLIPPER.LIB being specified last of all. The order that libraries are linked in is important, since libraries that use other libraries must be named first, and duplicate symbols are treated on a 'first come, first served' basis. Mutually dependant libraries may have to be linked with the `SEARCH` command to resolve cross references by making multiple passes. Third party libraries cannot always be `ALLOCATE`d (for example, *Tom Rettig's Library*), Blink Inc is negotiating with library manufacturers with a view to persuading them to produce Blinker versions.

Commands

Blinker script files can also contain commands that tailor the environment. Some of the most useful (stripped of their `BLINKER` prefix) are:

- `MINUTES`, `DATE` and `CALLS`, the three `DEMONSTRATION` command functions, cause a system to crash after a specified time interval, on a certain date, or after a certain number of overlay procedure calls. This allows you to provide potential customers with crippled versions of your applications.
- `EXECUTABLE SERIAL` encrypts up to 50 characters of serial number into the .EXE program, which is then available to the system and could be displayed to the user.
- `EXECUTABLE CLIPPER` allows the developer to burn-in the Clipper environment variable, instead of having to rely on schemes which rely on tampering with the punter's AUTOEXEC.BAT.
- `MEMORY PACK 15` causes the overlay manager to defragment memory after every 15 procedure calls, clearing up after macros etc.
- `OVERLAY OPSIZE` can be used to change the default overlay pool size (40KB), to reduce disk activity or increase free memory.

On test

The best way to explain Blinker is by a working example. I list below the stages I went through in 'Blinkering' an existing application. The program in question was already overlaid with PLINK86, the overlays having been balanced and optimised for size. Figure 5 shows a table of comparisons plus listings of two script files.

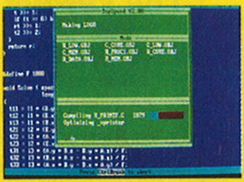
TopSpeed® C, Modula-2 and Pascal

Finally, a professional set of compilers designed to work together. All TopSpeed compilers share the same development environment and use the same optimizing code generator and run-time system.

	TopSpeed® C Version 1.00	Microsoft® C Version 5.1	Turbo C® Version 2.0	Watcom® C Version 7.0
100% ANSI compatible*	✓			
Integrated environment	✓		✓	
Pass parameters in registers	✓			✓
Expand any function as inline code	✓			
Supports OS/2	✓	✓		
DOS Dynamic Link Libraries *(overlay code linked at load-time)*	✓			
Smart linking *(only referenced code and data linked into .EXE)*	✓			
Type-safe linking *(function parameters and memory model checked at link-time)*	✓			
Fully automatic make works across libraries	✓			
Time-sliced scheduler for multi-tasking under DOS	✓			
Short pointers in any segment	✓			
Hypertext help with library online	✓		✓	

**Written by Neil Martin of the British Standards Institution (BSI) and printed in Personal Computer World June 1989, page 241.*

TopSpeed's seamlessly integrated multi-windowed environment.



VID (Visual Interactive Debugger): a source-level, multi-windowed symbolic debugger.

Standard Edition–
Optimizing DOS compiler, integrated source-level debugger, smart linker, automatic make, complete standard library, 6 memory models, BGI Interface, time-sliced scheduler, MS-DOS/BIOS interface, and more.

Extended Edition–
Standard Edition plus full source code to libraries, post-mortem debugging, MS Windows support, support for DOS DLLs (overlays), assembler, disassembler, profiler, DOS call monitor, and more.

OS/2 Edition–
Extended Edition (except for DOS compiler) plus ability to generate DOS executables, full support of OS/2 Kernel & PM calls, OS/2 smart linker, fully automatic generation of DLLs, source for protected mode libraries, and more.

Each compiler is available in three editions:

Standard Edition £149
(DOS Compiler & VID)
Extended Edition £295
OS/2 Edition £370

64-page Technical Specifications booklet describing TopSpeed compilers is available upon request.

ALL PRICES EXCLUDING VAT

To Order:

TEL: 071-253-4333

FAX: 071-251-1442

Call on shipping & handling charges & volume discounts. VISA/MC accepted.

30-day unconditional money-back guarantee.

In the U.S.A. contact:
Jensen & Partners Int.
1101 San Antonio Road
Suite 301
Mountain View
CA 94043
USA

Standard Edition $199
Extended Edition $399
OS/2 $495

Sieve (Msecs)	
TopSpeed V1.0	42.94
Watcom V7.0	44.31
Turbo V2.01	59.25
Microsoft V5.1	100.19

Plum Hall (secs)	
TopSpeed V1.0	1.41
Watcom V7.0	1.63
Turbo V2.01	1.84
Microsoft V5.1	3.79

Benchmarks measured by Mark Hamilton, November 24, 1989. Copyright © 1989, PC Business World and CW Communications Ltd.

TopSpeed®
C

Jensen & Partners UK Ltd.
Unit 21
63, Clerkenwell Road
London EC1M 5NP

TopSpeed and TechKit are trademarks of Jensen & Partners International. Other brand and product names are trademarks or registered trademarks of their respective holders.

CIRCLE NO. 145

The first test was to 'Blinker' the existing script file unchanged. The second stage was to use the `ALLOCATE` command to overlay EXTEND.LIB. The third stage involved rationalising the root module, and I decided to go the whole hog and wrote a new root module that contained a single command to call the menu system (`DO QM_MENU`) and moved everything else into the overlay area. This is just a lazy way out, since some of the old root code must always be active, and will therefore permanently occupy some of the overlay pool. The fourth stage involved compiling all the source code except the root module into a single .OBJ file, to save space in the symbol table. The final load size was down to 185 KB with the .EXE file less than 350 KB in size.

Running the resulting system proved that everything still worked, but there was much more disk activity as the overlay manager pulled overlays from the disk. To combat this, I increased the overlay pool size to 60 KB with the Blinker OVERLAY OPSIZE command. Sure enough, the disk loading overhead was reduced so that the final system was not noticeably slower than the PLINK86 version.

The finished product has a reported load size of 185 KB, or 245 KB including the 60 KB dynamic overlay pool, against PLINK86's 266 KB, which includes an 18 KB static overlay area. Blinker linked over 10 times faster than PLINK86, although some of the time made up was lost in increased compilation time producing a large OBJ file to reduce symbol space. The original PLINK86 overlays took about a day to balance and test. Converting to Blinker took a matter of minutes.

Figure 5 - Testing Blinker versus PLINK86.

Script file	PLINK86 1	Blinker 1	Blinker 2	3	4	5
Time (s)	28.12	2.86	2.74	2.69	2.63	2.63
EXE (bytes)	352496	357853	350341	361020	348396	348396
LOAD (KB)	266	252	212	196	185	185

Script file 1 (original PLINK86)

```
FI QM_MENU,QM_LIB
LIB CLIPPER,EXTEND,ASMLIB,OVERLAY
BEGINAREA
  SECTION FILE QM_GAN
  SECTION FILE QM_ADD,QM_CHK
  SECTION FILE QM_REP1,QM_REP2
  SECTION FILE QM_JDA,QM_ANA
  SECTION FILE QM_LST,QM_PROC,QM_BAN
ENDAREA
```

Script file 5 (final Blinker)

```
BLINKER OVERLAY OPSIZE 60        # Increase overlay pool to 60KB
FI QM                            # Root is just one command
LIB ASMLIB                       # Cannot ALLOCATE, interrupts
BEGINAREA
  FILE QM_ALL                    # All overlays in a single OBJ
  ALLOCATE EXTEND                # Overlay EXTEND library
ENDAREA
LIB CLIPPER
```

In profile

Blinker is supplied with a profiler, complete with full source code, that reports on the status of each overlay. The profiler can be invoked automatically each time an overlaid procedure is called, or manually through Clipper's DEBUG command or via a hotkey. The data provided by the supplied profiler consists of the name of the overlay procedure called, the time it was called, the amount of free memory (`MEMORY(0)`) after it was called, the number of times it has been called, the number of times it has been loaded from disk and the total size of all overlays currently loaded.The profiler can direct its output to a file, standard error, the printer or the screen.

Blinker provides a Programmer's interface consisting of some 25 functions. Eighteen report on some aspect of the current overlay status, the remainder allow the environment to be modified. Data available on demand includes a memory fragmentation table and the current status of demonstration settings, as well as access to the (decrypted) serial number and lots of information on the overlay pool and the procedures within it.

Functions that directly modify the environment include `BliOvlClr()`, which clears inactive overlays (freeing more Clipper pool before a memory-intensive activity), `BliMemPak()` which gets and/or sets the memory pack frequency.

`BliMemPack()` and its sister script file command, `MEMORY PACK`, overcome one of the most frustrating features of Clipper - its apparent lack of any garbage collection. There are a number of causes of memory fragmentation in Clipper: the use of macros and not closing files in the reverse of the order in which they were opened are two common ones. To test Blinker's defragmentation, I wrote a memory-junking routine that created a total of 1300 macros. Blinker's `BliDisFrg()` function, which displays a diagram representing memory fragmentation, was used to report on the state of things before and after a `BliMemPak(-1)`, which performs an immediate memory pack.

The results were impressive. True to form, memory was badly fragmented and stayed that way, even after a `CLEAR ALL`. `BliMemPak(-1)` could not quite get memory back to the condition it was in on start-up, but it came pretty close. `BliDisFrg()`, by the way, is rather clumsy to use, the table scrolling off the top of the screen at a rate of knots when a lot of memory is badly fragmented. A graph or some figures would be better, or even a straight percentage (which would allow the developer to pack memory when it had reached a certain level of fragmentation).

Blink Inc states that it is committed to expanding and enhancing Blinker. Planned over the next 12 months are support for Clipper 5.0, static overlays (for the masochist), incremental linking and storage of overlays in extended/expanded memory.

The Verdict

If you need overlays, Blinker is the one for you. It is worth bearing in mind, however, that a non-overlaid system will always be faster than an overlaid one, and you should still write tight code to avoid overlays wherever possible.

Like all great software, Blinker is based on a simple concept that has you wondering why it wasn't available years ago. A long overdue and very welcome relief for tired Clipper hacks: the next generation will wonder what all the fuss was about.

EXE

James Ormrod remembers Summer '85, and still bears the scars to prove it. He has just finished work on his 17th Clipper system and is a Nantucket Authorised Trainer. He can be contacted on (0491) 35187.

Blinker V1.11 is priced at £189.00 (free Clipper 5.0 upgrade to registered users) and is available from QBS Software Ltd. (0279) 501510.

The Evaluator

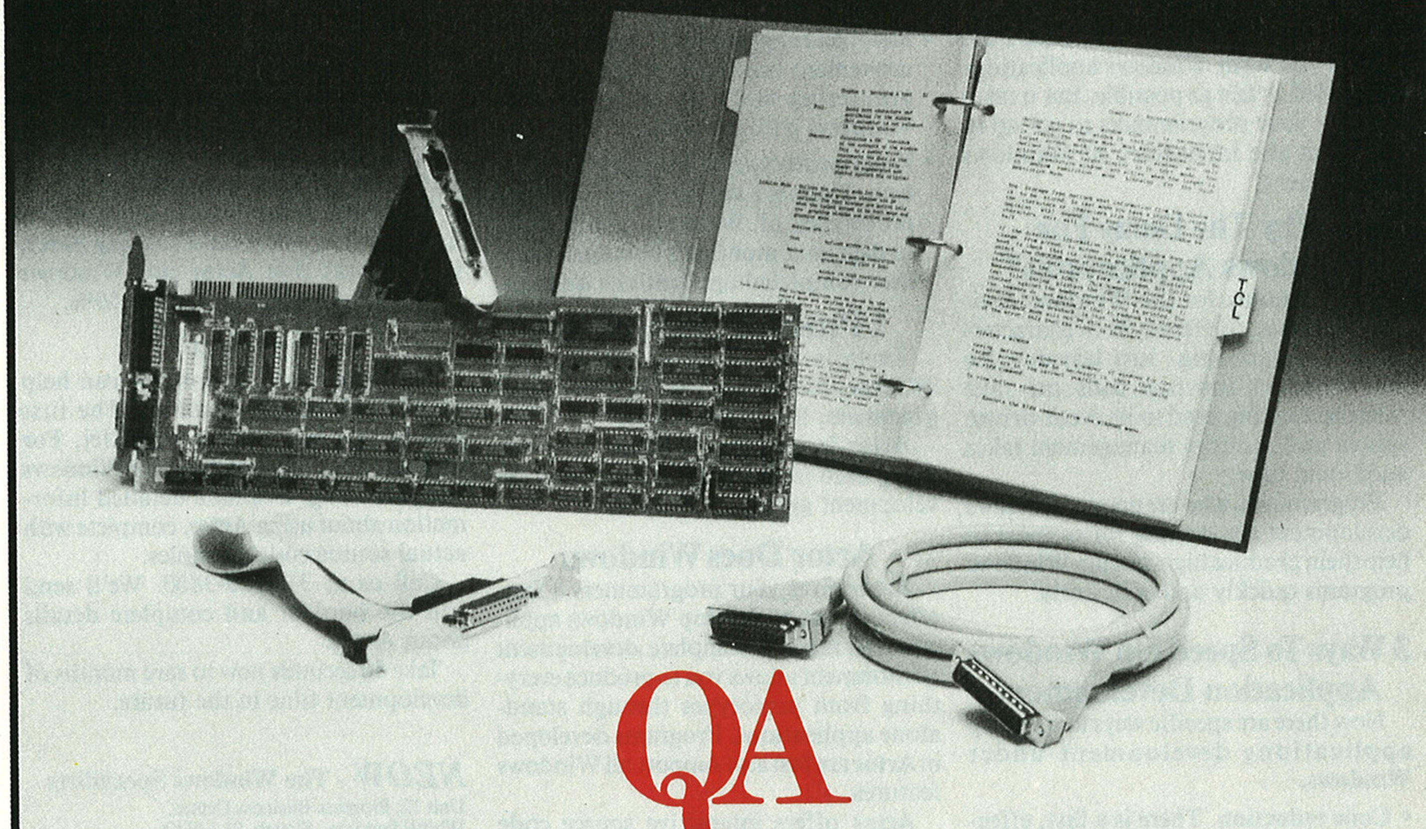
The Breakthrough in Software Testing

Completing the writing of your software is half the battle.

Ensuring that you have produced reliable fault free code is yet another major challenge. Testing software to be certain that full functionality is met during development or maintenance cycles is costly, time-consuming and prone to error...until now.

Evaluator, created by Elverex and supplied by QA, both simplifies and automates software testing, allowing you to create a comprehensive test process which fully exercises software to detect any errors. It saves effort, repetitive work, and frees the time of test personnel.

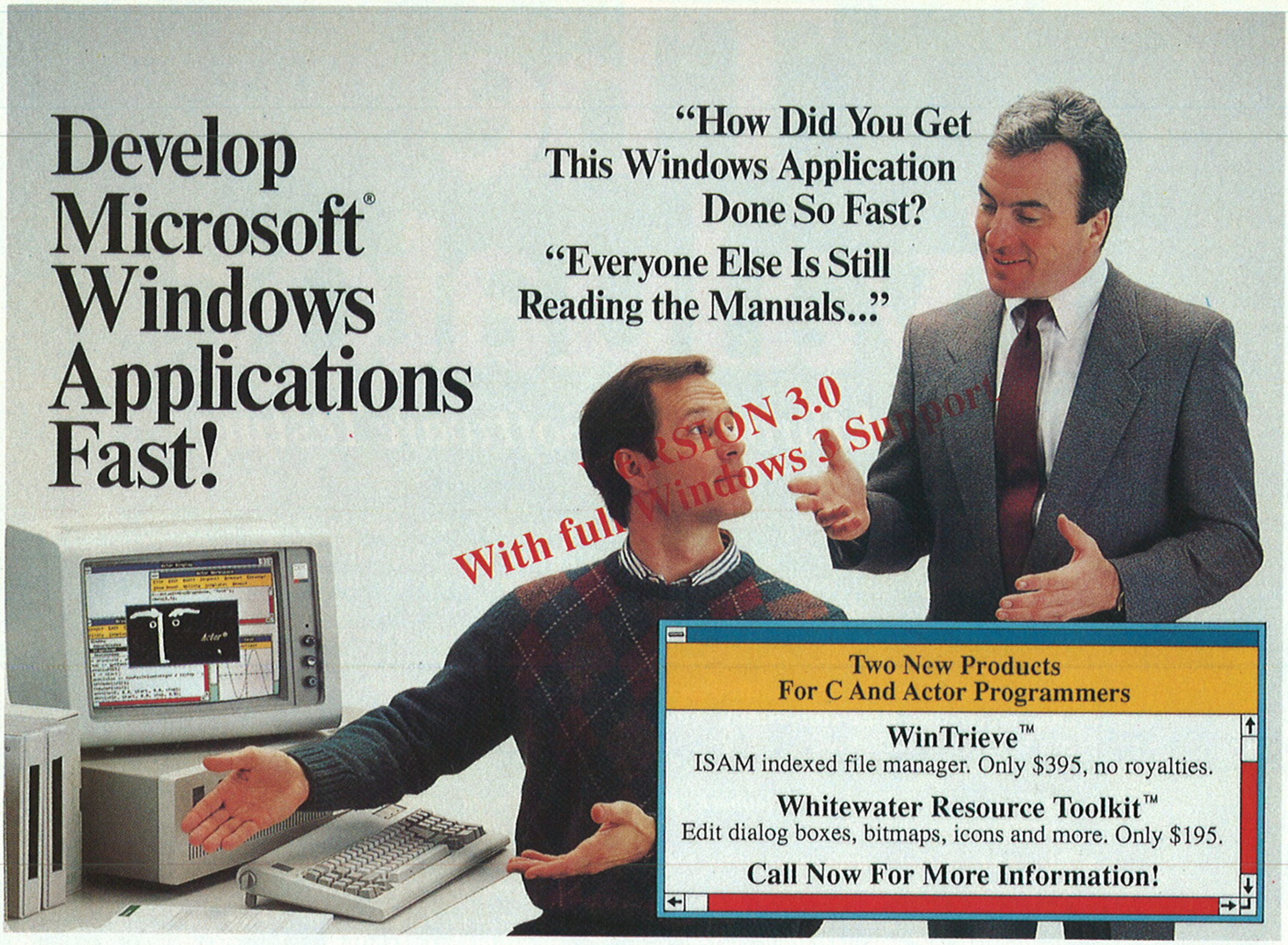
- *Interactive recording and playback of complex test sequences*
- *Rich test control language to write test programs*
- *Totally non-intrusive testing of any application on any operating system – DOS, Unix, OS/2, VMS, RMX...*
- *Automatic test report generation*
- *Unattended operation allows for more productive use of test personnel*
- *Easy to use control software for test management and monitoring*
- *Uses standard IBM PC or AT or 100% compatible as test station*
- *Tests software that runs on terminals or on a PC*



QA

Telephone us on (0285) 655888 for full information. QA, Cecily Hill Castle, Cirencester, Glos GL7 2EF

CIRCLE NO. 146



The Windows Advantage

You've selected Microsoft Windows as your graphical user interface. Windows applications are easier for end-users to learn and use. It is **the** standard for PCs.

Now you want Windows applications completed as fast as possible. But it takes time for your programmers to get up to speed on the intricacies of Windows development.

Why The Delay For Windows Applications?

Programming for the Windows environment is very different from character-based programming. Just learning the 450 Windows function calls can take months. Getting used to its event-driven system and memory management takes additional time.

Programmers who are new to Windows development are looking for support to help them produce high-quality Windows programs quickly and efficiently.

3 Ways To Speed Up Windows Application Development

Now there are specific ways to speed up applications development under Windows.

- **Code reduction.** There is a fast, effective way to develop Windows applications. A way that cuts the amount of code needed. A way that by-passes much of the complexity of Windows programming. A way that doesn't compromise performance.
- **Short learning curve.** There is an easy, convenient way to get your entire programming staff up to speed with Windows within weeks.
- **From prototype to complete application.** There is a way to quickly develop prototypes of Windows applications in days, not months. Then, turn those into completed applications quickly.

1 Product, 3 Solutions

There is **one** product which brings **all three** solutions to your Windows programmers. It's called **Actor®**

Actor is a proven product. It's now being used in thousands of software development groups world-wide.

Actor Does Windows

Actor gives your programmers a fast, effective way to develop Windows applications. **Actor's** complete development environment allows you to produce everything from prototypes through standalone applications. Programs developed in **Actor** are fast and support all Windows features.

Actor offers interactive source code debugging and testing—greatly reducing development time. Your programmers benefit from high code reusability because **Actor** is an object-oriented programming language. This allows them to build on existing code, eliminating duplication of effort. They can even dynamically link to existing C code.

Actor also provides one more essential: fast, responsive support from The Whitewater Group® recognized worldwide as leaders in Windows programming.

If you are programming applications for Windows, you should be using **Actor.** Users report that **Actor** cuts Windows programming time by at least 50%.

Act Now!

See for yourself how **Actor** can help your Windows programmers! The first step is to order our free booklet, **For Faster, Better, Smaller, Smarter Windows Programming.** Examine detailed information about using **Actor,** complete with actual source code examples.

Call us at 312/328-3800. We'll send you the booklet and complete details about **Actor.**

Take 60 seconds now to save months of development time in the future.

NEOW **- The Windows Specialists**
Unit 12, Progress Business Centre,
Whittle Parkway, Slough. SL1 6DQ.
Telephone: 06286 68334. Fax: 06286 65722

All trademarks acknowledged

CIRCLE NO. 147

It's not pronounced Squirrel

SQL is fast becoming the database standard. If you don't yet know how the relational model works, you will soon be dead meat: so you'd better read Jim Williams' explanation.

The 1980s was the hardware decade. In 1980, I started with 32 KB of RAM and 0.72 MB of disk storage in a Z80 North Star Horizon. By 1989, I had a 386 PC Compatible with 4 MB of RAM and 106 MB of disk storage, capable of running six jobs at once under DESQview. But as a developer of commercial applications, I am still writing the same programs: accounts, stock control, mailing lists and other systems for business users.

In this new decade, the continuing evolution of PC standards and the growth of networks emphasises the importance of connectivity of software applications. During the last few years commercial applications programmers have increasingly adopted proprietary database and 4GL products. The advantage of this approach is that reliable programs can be produced more cheaply than with the earlier 3GLs (such as BASIC and COBOL). The disadvantage is that applications become closed off to other software environments. A Sales Ledger written in dBASE would not easily combine with a stock control system written in Dataflex, except through the clumsy medium of import/export ASCII text files. When hardware dictated that applications were single-user, this drawback was not apparent. Now that PCs are being used as components of large networks, where the files of, say, the stock control department are required by accounts, this drawback becomes a serious problem.

Recognising this, the leading manufacturers are configuring their products to conform to the ANSI-defined standards of SQL (**S**tructured **Q**uery **L**anguage, generally pronounced 'sequel' not 'squirrel') and the look and feel of relational database architecture as defined by Ted Codd in 1970-72.

Database primer

There are three formally defined database models - Relational, Network and Hierarchical. The Hierarchical model is mostly of historic interest. It assumes that data may be organised in the classical directory/subdirectory structure and, if strictly followed, there can only be one forward link between a higher level of records and a lower level of records. This is not a good model of real world data. In practice, the hierarchical model tends to be compromised by duplication of data, plus some extra linkages between groups of records, to short-cut the normal search paths.

The Network model conforms better to the typical intuitive file structures that evolve in the real world. A group of invoice records may have multiple links - perhaps to the customer file and stock file - and in turn the customer file could link to the salesman/rep file. Each record type is a node, and the relationships are referred to as directed and non-directed arcs, depending on whether the linkage is one-to-one or one-to-many. The essence of the Network model is that only these two types of linkage are allowed *1->1* and *1->N*.

Relational Model

The most abstract model of database is the Relational model. The term *relational* is derived from the mathematical theory of relations which gives rise to 'relational algebra' and 'relational calculus'. The relational algebra covers the mathematical notation used to express the logic of relationships, calculus is the process of manipulating the expressions to derive more formal results.

As is usual with specialist disciplines, various terms often used in every day language have been poached and given precise meanings. An *entity* is an object with attributes. For example, an invoice is an entity, and its value is one of its attributes. Another attribute is the customer, which in turn itself has a set of attributes - name, address, credit limit and so forth. Where an attribute is an entity, then a *relationship* is said to exist. There are three types of relationship: one-to-one (*1->1*), one-to-many (*1->N*) and many-to-many (*m->n*).

A relation is depicted as a *table*, and each *column* of the table is an attribute:

```
Relation CUSTOMER:
NAME,ADDRESS,PHONE,BALANCE,SALES
MAN
```

This relation is often written as:

```
CUSTOMER(C#,NAME,ADDRESS,PHONE,B
ALANCE,SALESMAN).
```

The C# item is necessary because each *row* or *tuple* must be distinct, and must carry a unique identity which is the *primary key* - this is a stipulation of the relational model. The primary key may be artificial (such as an invoice's serial number) or natural - like a name that is known not to replicate. In a

```
1. CUSTomer = CUST(C_KEY,NAME,ADDRESS,TOWN,PHONE)
2. ITEMS bought = ITEMS(INVOICE_REF,STOCK_CODE,QTY,PRICE,C_KEY)

CUST:
[C_KEY]        [NAME]              [ADDRESS]        [TOWN]       [PHONE]
A1             ANDERSON TRADING    19 THE GROVE     CHISWICK     994-5124
B1             BARNS - SUPPLIES    20 HIGH STREET   ACTON        888-1234
C1             CARTER & SON        12 LOWER RD      ACTON        889-1234
C2             CRICKLEWOOD STORE   THE BROADWAY     CRICKLEWOOD  890-5467

ITEMS:
[INVOICE REF]  [STOCK CODE]        [DESCRIPTION]    [QTY]        [PRICE]  [C_KEY]
12333          WP1                 WHITE PAINT      10           2.90     A1
12333          BP1                 BLACK PAINT      5            3.20     A1
12333          CN                  COPPER NAILS     144          10.00    A1
12333          FB                  FLOOR BOARDS     8            16.20    A1
12334          WP1                 WHITE PAINT      8            2.32     B1
12334          BP1                 BLACK PAINT      4            2.56     B1
12335          CN                  COPPER NAILS     72           5.00     C1
12335          WP1                 WHITE PAINT      5            1.95     C1

NEW_CUST:
[C_KEY]        [NAME]              [ADDRESS]        [TOWN]       [PHONE]
A1             ANDERSON TRADING    19 THE GROVE     CHISWICK     994-5124
B2             BROWN - DIY         18 OLD STREET    NEASDEN      809-1299
C3             CARTWRIGHT BUILDER  10 DOWN LANE     EALING       778-1098
```

Figure 1 - Basic sales system.

Figure 2 - Who has not bought copper nails?

```
[C_KEY]        [NAME]               [ADDRESS]        [TOWN]        [PHONE]
B1             BARNS - SUPPLIES     20 HIGH STREET   ACTON         888-1234
C2             CRICKLEWOOD STORE    THE BROADWAY     CRICKLEWOOD   890-5467
```

table of car models, the term RENAULT 5 CAMPUS describes a particular model of car, and is unique to that make of car. Most primary keys are abbreviations derived from the name, such as RE5C.

Design

There are four stages of database design: *requirements analysis, entity relationships modelling, transformation to relations* and *normalisation of relations.*

Requirements analysis covers the stages of definition through to business systems design. In the customer sales example, at the end of requirements analysis one would expect to have identified the purpose of the system, its information content and its inputs and outputs.

Entity relationship modelling is taking the results of RA and defining the basic entities and relationships. In this case, our entities are `CUSTOMER`, `STOCK`, `INVOICE`, `ITEMS_SOLD` and `SALESMEN`. This is a crucial stage, as sound definition of entities determines how easy it is to produce a normalised final design (explained below).

Transformation to relations involves establishing the links between the newly-defined entities. We want to end up with a structure whose properties are dependent on the relationships. One property of the system is that it must print out invoices, which implies the following relationship:

```
INVOICES->CUSTOMER
  ->ITEMS_SOLD
```

We also need to report which invoices are outstanding for each customer, and what each customer has purchased:

```
CUSTOMER -> INVOICES
   -> ITEMS_SOLD
```

Each property of the system must be tested against the relationships, to establish whether the model supports the user requirements.

The above three stages may be repeated until the designers are satisfied that the model meets the functional requirement.

We are normal

Normalisation removes ambiguities from the model. There are various levels of stringency of normalisation, known as normal forms.

First Normal Form is achieved when all entities cannot be decomposed further - all attributes within a relationship are single 'atomic items'. This would debar a structure where the details of goods sold (invoice lines) and the overall accounting information such as total, due date, and salesman code are held in the same 'invoice' table. Because the sales may be decomposed to invoice lines, the invoice lines must reside in a separate table. Second Normal Form demands First Normal Form (1NF) as a prerequisite. 2NF is reached when non-prime attributes are decomposed to key fields. In our sales accounts example, the invoice might store delivery addresses. However, these delivery addresses could repeat in a sequence of invoices to the same customer. In Second Normal Form, a new table for delivery addresses is established, and the invoice table will have a field defined as the key for the delivery address.

There are up to five normal forms for relational decomposition, but in practice we need only concern ourselves with one more. Third Normal Form is the elimination of secondary dependencies with a relationship. If we stored details about the customer's credit limit and current balance in the invoice header relationship, we have a secondary dependency. This is because credit limit belongs to the customer, and is not generally related to invoice. The credit limit must be moved to a customer file (or a new table could be set up, storing the credit information for each customer).

The Relational Model & SQL

The Relational Model is based on precise notation and definition, which is also applied to the syntax of SQL. Each command in SQL mirrors an operation in relational calculus. It is this theoretical underpinning that makes possible complex extractions of information from a database using a sparse vocabulary. The following concepts from relational algebra and calculus are applicable is SQL.

Set Operation: Given tables ('A' and 'B') with similar attributes, the following set operations apply:
Union: All members of A and B eliminating duplicates. Intersection: Members of both tables.
Minus: A - B = Members of A not in B.
Selection: Select from a table those rows which satisfy one or more criteria.
Projection: Extract a set of columns from a table eliminating duplicate rows. In our sales example, the projection of customers from invoices would provide a column of customers with duplicates. The final result would not include second and subsequent occurrences of a customer ID.
Cartesian Product: The result of `A x B` ('A times B') would be a table showing all possible combinations of tuples.
Natural Join: Two tables are linked by a common attribute. A list of all invoices and customer details may be produced by a join between the customer file and invoice file, using the attribute 'customer code' as the common attribute.

SQL has the syntax for all the above five operations of relational algebra. As a result, sophisticated queries can be formulated in a surprisingly succinct manner. We shall consider the case of our sales system analysing sales data. We have two main tables, `CUST` and `ITEMS`, illustrated in Figure 1. Note these are not 2NF, as `DESCRIPTION` should live in a `STOCK CODES` file. It is included here for 'readability'. We also have a third, auxiliary file called `NEW_CUST`, also shown in Figure 1.

The core of SQL is based on the command `SELECT`, which displays attributes from a table. It takes two modifiers:

`FROM` (which defines the tables to use) and `WHERE` (specifies the conditions of selection). The results of each `SELECT` query are displayed in tabular format. `SELECT *` displays all attributes. Thus

```
SELECT *
 FROM ITEMS;
```

```
SELECT CUST.NAME, CUST.PHONE, ITEMS.DESC
FROM CUST, ITEMS
WHERE CUST.C_KEY = ITEMS.C_KEY AND
(ITEMS.STOCK_CODE = "WP1" OR ITEMS.STOCK_CODE = "BP1");

NAME                  PHONE          DESC
----                  -----          ----
ANDERSON TRADING      994-5124       WHITE PAINT
ANDERSON TRADING      994-5124       BLACK PAINT
BARNES SUPPLIES       888-1234       WHITE PAINT
BARNES SUPPLIES       888-1234       BLACK PAINT
CARTER AND SON        889 1256       WHITE PAINT

5 rows selected
```

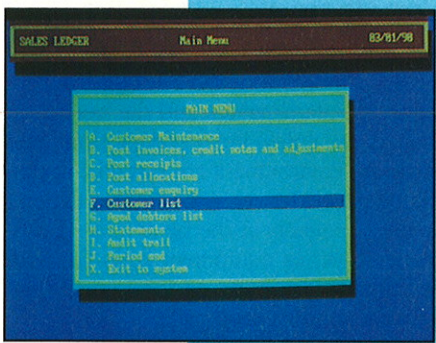
Figure 3 - Creating a view.

dBASE USERS!

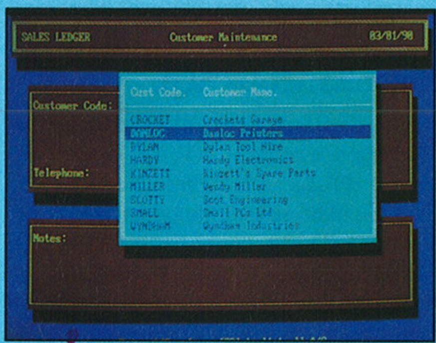
How many days would it take to program, test and debug this system? With Sycero and Clipper you could do it in HOURS!

Clipper is the applications development standard for PC database management systems. Clipper's open architecture and superior networking capabilities let you create more sophisticated and powerful applications.

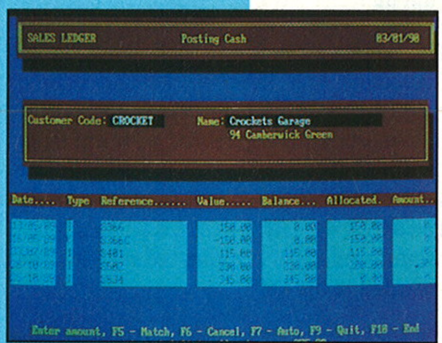
Sycero is a complete applications generator for Clipper. Making use of all of Clipper's advanced facilities, Sycero lets you develop PC applications in a fraction of the time it would take to do manually.



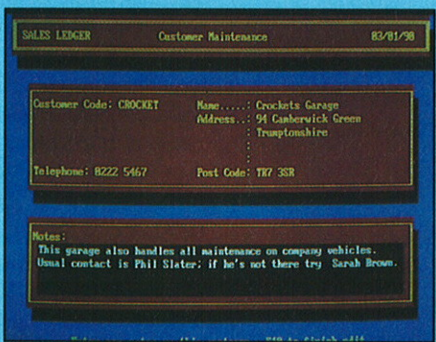
Pulldown, bouncebar and Novell style menus produced in minutes.



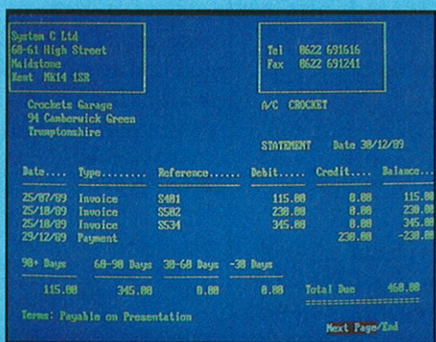
Overlay a browse window with a function key in just five lines of code.



Automated transaction handling.



Memo fields give you word processing in a window on your screens.



Complete & powerful report generator for forms, lists and more complex invoices/statements.

Sycero with Clipper – the perfect development environment.

Please send me further details about Sycero & Clipper:

Name: ....

Address: ....

....

....

Tel: .... Fax: ....

System C Ltd., 60-61 High Street, Maidstone, Kent ME14 1SR

Tel:0622 691616 Fax:0622 691241

.EXE 7/90

Nantucket

System C

CIRCLE NO. 148

dBASE® IS A REGISTERED TRADE MARK OF ASHTON-TATE.

displays all the records in `ITEMS`, and

```
SELECT ITEM.DESCRIPTION,
  ITEM.C_KEY
 FROM ITEMS
 WHERE ITEMS.C_KEY = "A1";
```

displays a table of all item descriptions and customer code where the customer key is A1. The syntax `TABLENAME.ATTRIBUTENAME` is common to most relational syntax.

Suppose we wish to have a 'join' to the customer table, to obtain the name of the customer, instead of just the key.

```
SELECT ITEM.DESCRIPTION,
  ITEM.C_KEY,
  CUST.NAME
 FROM ITEMS , CUST
 WHERE ITEMS.C_KEY = "A1"
 AND ITEMS.C_KEY = CUST.C_KEY;
```

One of the underlying mechanisms of SQL is the production of tables. Queries can be nested:

```
SELECT *
FROM CUST
WHERE C_KEY NOT IN
 (SELECT C_KEY
 FROM ITEMS
 WHERE
ITEMS.STOCK_CODE = "CN");
```

The result of the above query will list all Customers who have not bought copper nails (Figure 2). Note the use of `NOT IN` in this query. The script starts with the inner loop, creating a list of `C_KEY`s in the `ITEMS` file of purchases of copper nails. The `WHERE C_KEY NOT IN` scans this list, printing all `CUST` tuples which are not included in the list.

Qualifiers

If we wished to list all products sold at least once, then

```
SELECT DISTINCT STOCK_CODE
 FROM ITEMS;
```

would produce the following table:

```
STOCK_CODE
----------
WP1
BP1
CN
FB
```

The `DISTINCT` qualifier eliminates duplicate occurrences from the table. The `GROUP` modifier aggregates numeric fields, based on the attribute chosen for the `GROUP` clause.

```
SELECT STOCK_CODE SUM (PRICE)
FROM ITEMS
GROUP BY STOCK_CODE;
```

produces the following result:

```
STOCK_CODE       SUM(PRICE)
----------       ----------
BP1              5.76
CN               15.00
FB               16.20
WP1              7.17
```

`ORDER BY` sorts a particular column.

```
SELECT NAME PHONE
FROM CUST
ORDER BY NAME DESCENDING;
```

produces:

```
NAME                   PHONE
----                   -----
CRICKLEWOOD STORES     890-5467
CARTER AND SON         889 1256
BARNES SUPPLIES        888-1234
ANDERSON TRADING       994-5124
4 rows selected
```

Note how the descending modifier reverses the order.

An important feature of relational theory is the ability to join two tables to produce a secondary table. This derived table is described as a *view*, and given its own distinct name. Further selections may be made from the view to produce a report which combines information from two or more base tables. Figure 3 illustrates this approach, to obtain the names and phone numbers of all the customers who have bought paint.

Miscellaneous SQL

`WHERE` clauses may be constructed using the following operators: =, <>, !=, >, <, >=, <=, `IS NULL`, `IS NOT NULL` and `LIKE`. The right hand side of the operator may be either a constant or column name. Operations may be combined by means of `AND` and `OR` statements.

Figure 4 shows some other operations which can be carried out on (generally numeric-only) columns.

Figure 4 - Column operators.

Operator	Function
AVG ({DISTINCT,} COLUMN.NAME)	Mean
SUM {DISTINCT,} COLUMN.NAME	Total
COUNT {DISTINCT} COLUMN.NAME	Number of Rows
MIN (COLUMN.NAME)	Minimum Value
MAX (COLUMN.NAME)	Maximum Value

Naturally, SQL provides the means to create and update tables (although for some reason, this fundamental operation always appears towards the end of SQL tutorials). To create a table:

```
CREATE CUST (
 CODE char 2
 NAME char 20
 ADDRESS char 40
 TOWN char 20
 PHONE char 8
 ) 20;
```

where the figure following the closing bracket stipulates the number of rows to create.

The `STRUCT` command, used:

```
STRUCT CUST;
```

lists out the format established by `CREATE`. `DELETE` is used to delete either whole tables, or just selected rows specified by a `WHERE` clause. So

```
DELETE ITEMS;
```

deletes the whole `ITEMS` table, but

```
DELETE ITEMS
 WHERE
 ITEMS.STOCK_CODE = "BP1";
```

selectively deletes all rows with sales of black paint, leaving a smaller table.

`INSERT` adds extra rows to a table, and `UPDATE` changes selected rows and columns:

```
UPDATE CUST.PHONE
FROM CUST
WHERE CUST.TOWN = "ACTON";
```

affects all fields defined by the intersection of `PHONE` and `CUST.TOWN = "ACTON"` in table `CUST`. These are displayed in turn and replaced by keyboard entries. Note that the facilities for modifying tables vary slightly between implementations of SQL. Versions designed for the end user, who may not be authorised to access, say, a corporate database, may not offer any update facilities at all.

Conclusion

It is hard to understate importance of the relational model and SQL. The conceptualisation of data as tables and the notion of data as an entity independent from the hardware environment are essential in developing tools for rapid data manipulation. As my comments at the beginning of this article show, as equipment costs fall and performance increases, it is essential that software vendors keep pace.

For example, one of my clients, the MD of a medium sized company, regularly downloads his whole company database - 2,000 customers, two years sales ledger, three years back sales history and a 1000 item stock file - onto his PC. He uses this data to prepare various reports and analyses that enable him to construct his business plans from home, or even while travelling abroad. With a suitable SQL system - I use Dataflex's FLEX-QL, because I program with that company's proprietary database language products - he can treat his corporate database as a massive spreadsheet which updated once a week. He uses a copy of the same database that is directly updated as orders are entered, goods sold and new customers acquired. There is no intermediary between the real facts of his business and his need for information to control the future.

EXE

Jim Williams is an Independent Software Consultant, specialising in business applications mainly written in Dataflex. He has been programming micros since 1975. He can be contacted on 081 892 0538.

QBS Software Limited

Specialising in the best add-on software for the Clipper developer.

QBS Software Limited distributes throughout Europe and Scandinavia, providing fast delivery and comprehensive technical support. We are currently selling and supporting **FUNCky**, a library of over 400 functions and **BLINKER**, the dynamic overlay linker for Clipper. Our product range is expanding to cover graphics, communications, networking and other important areas. All our software will be compatible with Clipper 5.0, shortly after the new compiler is available.

For details of our complete range of products and how they can help you create more exciting applications, please contact:-

FUNCky

QBS Software Limited
15, Twyford Business Centre
Twyford Road
Bishops Stortford
Herts CM23 3YT
United Kingdom
Telephone: 0279 501510
Fax: 0279 501545

BLINKER

Clipper is a registered trademark of Nantucket, Blinker is a registered trademark of Blink Inc., FUNCky is a registered trademark of dLesco Associates.

CIRCLE NO. 149

People are talking about us.

F77L-EM/32

Port 4GB mainframe applications to 80386s with this 32-bit DOS-Extender compiler. The winner of PC Magazines's 1988 Technical Excellence Award just got better. New Version 3.0 and OS include: Editor, Make Utility, Virtual Memory Support, DESQview Support, New Documentation and Free Unlimited Runtime Licenses. F77L-EM/32 with OS/386 £895.00 (+ VAT)

F77L

The compiler of choice among reviewers and professionals. Includes a Debugger, Editor, Profiler, Linker, Make Utility, Weitek and 386 Real-Mode Support, Graphics. £395.00 (+ VAT)

Lahey Personal Fortran 77

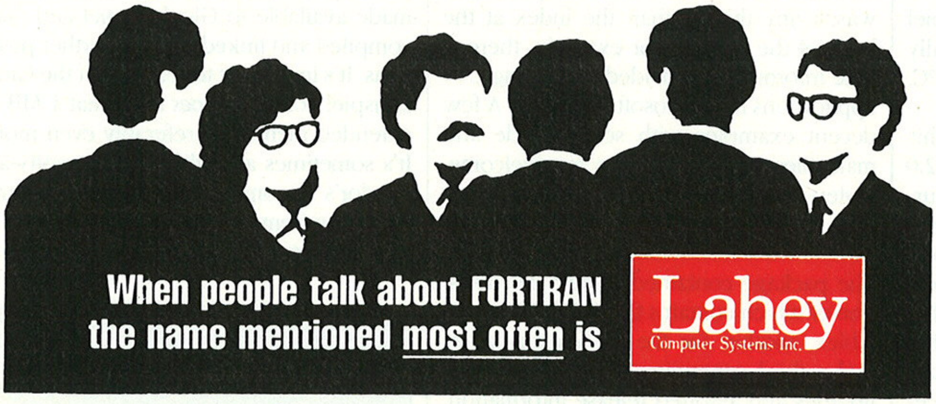
New Version 3.0: Full ANSI 77, Debugger, Editor, Linker, Library Manager, Microsoft and Borland C interfaces, 400 page Manual. Unbeatable Price £75.00 (+ VAT)

NEW RELEASE

F77L-EM/32 includes:
- true 32 bit code
- 80387 and Weitek support
- VAX/IBM extensions
- New Editor
- Screen Graphics
- Linker
- Make Utility
- All new documentation

With Lahey ERGO OS/386:

- Virtual Memory Manager for HUGE programs
- Unlimited Run-time distribution



System Science

3-5 Cynthia Street,
London N1 9JF
Tel: (071) 833 1022
Fax: 081 837 6411

CIRCLE NO. 150

Last of the preprocessors: Glockenspiel C++ 2.0

Dave Jewell checks out Glockenspiel's latest C++ offering and decides that all is not what it seems....

Although 'the Year that Unix Really Arrived' never seems to get here, 1990 may justly claim to be the year when C++ finally moved into the mainstream PC programming arena. Among PC programmers, Turbo C has always had a large and devoted following; the recent introduction of Turbo C++ has provided an easy upgrade path for those keen to begin exploring this new language. Last year saw the introduction of Zortech C++ 2.0 and, as I write, Zortech are about to release version 2.1 of their compiler which implements a number of previously left out features of the language and also provides support for Windows 3.

Both Zortech and Turbo C++ are native compilers. They take C++ source files as input and produce standard Microsoft format object files as output. But the first C++ compilers to arrive on the PC scene were translators. C++ translators produce C source code which then needs to be compiled using a standard C compiler. I believe I'm right in stating that the Glockenspiel C++ translator was the first commercially available implementation of C++ for the PC.

Glockenspiel C++ 2.0a, the subject of this review, is the company's latest C++ 2.0 compiler. Glockenspiel (like all other current C++ translators) is based on the standard translator developed by AT&T: *Cfront*. Version 2.0 of Cfront adds a number of new features to the C++ language. Most notable among these additions are: multiple inheritance, type-safe linkage and virtual base classes. As well as the version 2.0 enhancements to the language itself, Glockenspiel now supports the new Microsoft C 6.0 compiler and the associated PWB (Programmer's WorkBench) environment. An online help reference is also supplied. More on these later. Glockenspiel also supplies *CommonView*, a set of C++ class libraries for programming under Microsoft Windows and OS/2 PM. This may be ordered separately from Glockenspiel, and does not form a part of this review. *(From the beginning of August, Glockenspiel will bundle its C++ compiler and CommonView as one package - Ed).*

Opening the box...

The Glockenspiel documentation is contained within a single three ring binder and is divided into four parts, these being: *User Guide, Compiler Guide, Syntax Manual* and *Library Guide*. The latter two are of particular interest, since they are essentially reprints of the definitive AT&T manuals. The other documentation contains a comprehensive list of available compiler switches and other options, although I felt that the information provided here was rather minimal. The User Guide in particular wasn't any thicker than the index at the back of the binder. For example, there is little information provided on writing C++ applications for Microsoft Windows. A few decent examples with source code and make files would have been very welcome. A step-by-step tutorial would be even better.

The package contained three 5.25" disks, one of which contains the MS-DOS executables, another for the OS/2 executables and a third containing common headers, libraries and assorted release information. A set of 3.5" disks was also supplied. The installation program asked the usual sort of questions: where it should put the libraries? header files? and so on. It then installed the system painlessly. Disk space requirements are quite modest :- the MS-DOS system required less than 2 MB.

Memory Requirements

After installing Glockenspiel onto my hard disk, my intention was to begin by compiling the classic 'hello' C++ program. This masterpiece is illustrated in its entirety in Figure 1.

Unfortunately , my initial attempts to compile this program resulted in the compiler complaining about a lack of memory. Glockenspiel states that its compiler needs to run on an 80286 or 80386 system with 640 KB of base memory and a minimum of 1 MB of extended memory. Unfortunately, when I'm not running Windows, the extended memory on my machine is normally configured as a RAM cache and consequently it wasn't available to the compiler. After editing my CONFIG.SYS file and rebooting, all my extended memory was made available to Glockenspiel and *hello* compiled and linked without further problems. It's important to realise that the Glockenspiel compiler *does* need that 1 MB of extended memory, preferably even more. It's sometimes all right to take a software vendor's 'recommended' minimum hardware requirements with a pinch of salt. In this case, you shouldn't.

```
#include "iostream.hxx"

void main (void)
{
  cout << "Hello Glockenspiel....";
}
```

Figure 1 - C++ Hello program.

While on the subject of memory usage, it's interesting to note that the compiler is implemented using the Rational Systems DOS/16M Extender, and it's through this go-between that it makes use of extended memory. I've never quite figured out why it is that a typical port of the AT&T Cfront translator seems to require even more memory than a native code C++ compiler implementation, but Glockenspiel C++2.0 is no exception. Sadly, you can't make use of the Rational Extender technology from within your own software :- if you want to do that you have to buy the DOS Extender for yourself. In this respect, Borland's Turbo C++ VROOM system comes out on top because VROOM allows you to compile bigger programs (it's built into the compiler) *and* it can be used inside your own software at no extra cost. Although the new Zortech 2.1 C++ system also makes use of the same Rational Systems Extender, Zortech provide their own VCM (Virtual Code Management) technology for inclusion in your own programs. The bottom line is: when it comes to writing *big* MS-DOS based C++ programs, Borland and Zortech have done all the hard work for you; with Glockenspiel it's an optional extra.

The compiler in use

The Glockenspiel compiler is no speed demon. It chugs along happily at the sort of speed that is all too familiar to Microsoft C users. Like Microsoft, it takes a large and sometimes bewildering range of compiler switches. For convenience, commonly used compilation switches can be set up in an environment variable called **CCXX**. CCXX.EXE is also the name of the compiler 'driver' :- the small program which executes the various parts of the compiler proper, and then invokes the Microsoft C compiler and the linker to produce the finished executable program.

Broadly speaking, the command line switches fall into one of three categories: those beginning with a '!' character control the operation of the CCXX compiler driver program. These switches offer a useful range of options, including the ability to generate a list of commands which can be redirected to a batch file. For machines without much memory, this batch file can then be executed directly without the overhead of having the compiler driver in memory also.

Glockenspiel chugs along happily at the sort of speed that is all too familiar to Microsoft C users

The second category of switches is preceded with a '+' character. These switches control the operation of the C++ compiler itself. An interesting option here is the ability to generate code compatible with Microsoft C 5.1. (By default, generated code is compatible with Microsoft C 6.0.)

The final switch category starts with a '-' character. These switches control various aspects of the preprocessing and linking phases. Unrecognised switches in this category are passed through to the native code compiler. For example, you could specify '-Gsw' to instruct Microsoft C to generate Windows compatible code.

A very big surprise for me was the lack of support for anything except the large memory model. Maybe I'm missing something fundamental. I really can't understand the need for this restriction. The Glockenspiel compiler is quite capable of producing small and medium model code :- I tried a few experiments to prove it. The problem is that *only* large model libraries are provided. A total of six different large model libraries are supplied covering the various combinations of calling convention (stack-based and register-based) and 8087 support (8087 mode, emulation and alternate). Given the choice, I suspect that many potential customers users would prefer having fewer large model libraries and more of the other models! By contrast, both Zortech and Borland provide the usual range of memory models.

The Glockenspiel compiler produces C source code files with an extension of '.I'. These files are then processed by the Microsoft C compiler to produce object code. I wouldn't recommend browsing through one of these C source files unless you have a strong stomach :- they're not intended for human consumption! This isn't a reflection on the Glockenspiel implementation. Any version of Cfront produces similar output. The C code looks particularly tangled if you include and use the IOSTREAM.H header file, which contains the C++ interface for the new AT&T stream library. This is because AT&T seem to have a tendency to use `inline` member functions with great enthusiasm. The code in the IOSTREAM.HXX header contains `inline` functions which call `inline` functions which call....well, you get the idea. The example *single* C statement in Figure 2 should give you a flavour of what can happen.

Supplied Utilities

As a part of the Glockenspiel C++ package, Glockenspiel supply what looks like a complete copy of the previously mentioned *Syntax Manual* as an online reference. The help file, called CXXSYN.HLP, is a help database file in the same format as that used by both the Microsoft QuickHelp utility and the PWB help facility. Just in case you don't have the QuickHelp program (ie you are a C V5.1 owner), Glockenspiel has supplied a copy as part of the package.

Another interesting utility is a program called CXXFILT. This is a C++ name 'unmangler' which restores mangled C++ names back to a human-readable form. For those unfamiliar with the concept of C++ name mangling, it involves encoding the number and types of a function's parameters into the function name. All C++ 2.0 compilers implement 'type-safe linkage', whereby the linker is able to check that all references to a function across all object modules agree with the actual function declaration. This is achieved by name mangling, which has the big benefit that a standard linker can be used. The linker will complain about unresolved refer-

```
if ( ( ( ( __0this -> Pios-> bp__3ios -> x_pptr__9streambuf >= __0this ->
Pios-> bp__3ios -> x_epptr__9streambuf ) ?( ( * ( ( ( int ( *
) ( struct streambuf * __0this , int __1c ) ) ( __0this -> Pios->
bp__3ios -> __vptr__9streambuf [ 2] ) .f) ) ) ) ( ( ( struct
streambuf * ) ( ( ( ( char * ) __0this -> Pios-> bp__3ios )
) + ( __0this -> Pios-> bp__3ios -> __vptr__9streambuf [ 2] ) .d) )
, ((int )__0c )& 0377 ) :( ( ( ( * ( __0this -> Pios->
bp__3ios -> x_pptr__9streambuf ++ ) ) ) = ((int )__0c )) & 0377 ) )
) == -1) {
( ( ( ((struct ios * )__0this -> Pios)-> state__3ios |= ( 3& 0377 ) )
, ( ( ((struct ios * )__0this -> Pios)-> ispecial__3ios |= ( 3& ( ~
512) ) ) , ( ((struct ios * )__0this -> Pios)-> ispecial__3ios |= 3) )
) , ( ( ( char ) 0 ) ) ) ;
```

Figure 2 - A single C statement output by Glockenspiel.

ences if a type-mismatch occurs between modules. The CXXFILT program is implemented as a filter taking its input from 'stdin' and writing the results to 'stdout'. It can be used to unmangle unresolved references generated by the linker as an aid to determining what type mismatch errors have occurred. I found that, although CXXFILT worked most of the time, there were some mangled names that it refused to unmangle. For example, it had problems with constructors, destructors and with overloaded operators.

For those who like using the Microsoft PWB environment, Glockenspiel includes a PWB extension which integrates C++ into the PWB system. Files with an extension of CXX are recognised as being C++ files and the Glockenspiel compiler is automatically invoked to build the corresponding object files. Menu choices allow you to select either 'C++ Debug Options' or 'C++ Release Options' for controlling the insertion of debugging information into the object code. Support for the PWB Source Browser is also provided although for maximum readability you would have to use the !M compiler switch which disables inline function generation and minimises name mangling. The Browser doesn't understand C++ member function names.

Stop Press

Since Dave finished his review, Glockenspiel has upgraded its C++ compiler to V2.0c. We have not seen the product ourselves, but the manufacturer reports a variety of improvements. Addressing one of the criticisms of this article, a new filter program called FXX has been introduced. This processes the debug information in an executable file, 'transforming mangled C back into C++'. CodeView users can now see the correct C++ code in their local, call stack and watch windows. The PWB browser also now becomes usable at C++ source level.

Other enhancements: there is now support for creating DLLs, positioning of PWB's editor at the site of compilation errors, 31 character identifiers are allowed, 64 KB character arrays can be produced (limit up from 32 KB) plus various other tweaks and fixes.

Probably one of the most important reasons for considering Glockenspiel's C++ implementation is the tie-in with the CommonView class libraries, so it is relevant to note that a new version of CommonView, for Microsoft Windows V3.0, will be launched around the same time that this issue of .EXE is published. *Ed.*

Code Generation

Commenting on the object code quality of a C++ translator is somewhat irrelevant, since the native C compiler (in this case Microsoft C 6.0) is the real producer of the object code. Having said that, there are in fact plenty of optimisation opportunities which are available to Cfront itself and I've yet to see a version of the translator which avails itself of many of them. For example, how often can you see the construct `__0this -> Pios-> bp__3ios` in the code fragment in Figure 2?

I wouldn't recommend browsing through these C source files unless you have a strong stomach

Bearing in mind that Glockenspiel restrict users to the large memory model, this sort of thing is going to eat up a lot of code unless it's used in conjunction with a fiendishly cunning optimiser in the native C compiler. Microsoft C 6.0 isn't that cunning. Ideally, the C++ translator should set up a temporary pointer variable and use this to reference deeply indirected class members. This would be analogous to a hidden `WITH` clause in the Pascal language. Such optimisations would drastically improve the quality of the final code (both in terms of increased speed and reduced size).

Of course, major changes to Cfront such as the one I've described aren't really Glockenspiel's responsibility. AT&T are the developers of Cfront :- it's up to them to ensure that reasonable quality code will be produced even when used with the stupidest of non-optimising compilers.

Windows Support

Because you can pass command-line switches through to the Microsoft C compiler, it's easy with Glockenspiel C++ to generate code in the format needed for use under Microsoft Windows. Because the final code is produced by the new Microsoft compiler and linked with the new Microsoft Windows libraries, there are no problems regarding compatibility with Windows 3. The restriction of large model only libraries could prove to be a problem, however. The majority of non-trivial Windows programs are written to use the medium memory model.

Conformance to V2.0 C++

A major advantage of a C++ translator based on AT&T's Cfront program is the fact that it's based on an absolutely standard implementation of C++. Zortech's early C++ offerings were notoriously deficient in certain areas (the latest Zortech compiler, version 2.1, will be the first to add pointers to members) and you would be unlikely to encounter this sort of problem with Glockenspiel C++. I tried out a number of C++ 'torture tests' on the compiler and didn't encounter any problems.

Conclusions

Although suppliers of C++ translators such as Glockenspiel, Comeau and Intek, (and perhaps AT&T themselves) would undoubtedly disagree with me, I strongly feel that the days of C++ translators are now numbered, at least as far as the PC market is concerned. Zortech hammered the first nail into their coffin; Borland recently administered an even bigger nail; and Microsoft (when they eventually get round to it) will very likely supply the biggest nail of all in the shape of a native code C++ compiler and development environment which seamlessly integrates MS-DOS, Windows and OS/2 development. So what are my reasons for taking such a heretical stance?

Originally, Bjarne Stroustrup and his colleagues at AT&T took the decision to write Cfront as a translator rather than as a native code compiler for reasons of portability. It meant that C++ could be implemented relatively easily on any platform which already had a workable C compiler. Now that there are at least two native code C++ compilers for the Intel 80x86 microprocessor family, implementation portability is clearly less of an issue. Secondly, as I have already pointed out, the intermediate C code generated by a Cfront translator leaves a lot to be desired in efficiency terms. Even a reasonably good code optimiser such as that in Microsoft C is unable to take a 'higher-level' view and implement the sort of pointer optimisations I've mentioned. Lastly, the presence of a third-party native C compiler between the C++ translator and the

Asynchronous Point-to-Point Communications or Telephone-polling Multiple Computers

XOREN COMMS SOFTWARE– Software with ten years' pedigree

Thoughout the 1980s, Xoren Computing pioneered the development of inexpensive yet reliable communications software. Today, the XOREN IPL-11 range is renowned for enabling automated file transfer between any quantity and combination of the major minis, micros and PCs in configurations which will become more and more commonplace through the 1990s.

Leading with Automated Connectivity

Although many other communications packages can asynchronously link PCs together, very few besides IPL-11 will orchestrate *fully automated connectivity* between multiple PCs and hosts.

Automated operation with IPL-11 gives users the freedom to perform unattended data communications 24 hours a day. Computers with modems operating up to 9600 baud can be scheduled to dial up remote computers over ordinary telephone lines day or night to send and receive files. Any IPL-11 version can link and transfer files to any other IPL-11 version, regardless of the computers' operating system. Once the files have arrived safely, telephone lines are disengaged.

Leading with Customer Support

Key to the success of the IPL-11 range is an emphasis by Xoren on customer support.

Because communications is notorious for being a difficult application to implement, Xoren offers installation and support services to ensure the software can operate according to the *user's* requirements.

This principle has made IPL-11 communications software the obvious choice for UK and European companies. Because IPL-11 is a British product, developed and supported in the UK, Xoren is strategically placed to support customers when they need support most - during regular business hours.

Call or write to Xoren today - the most practical solution to automated data communications has always been right on your doorstep.

XOREN IPL-11 Developments over a Decade

1980: Xoren launches IPL-11. Features include peer-to-peer file transfer under operator control, CRC error-checking on individual packets and queuing of file transfer commands. *Original versions introduced for RSX-11M and VMS for DEC PDP-11 and VAX.*

1981: Remote activation facility incorporated allowing file transfers when remote computers are unmanned. *RSTS/E, RT-11 and RSX-11M PLUS versions released for DEC PDP-11.*

1982: Terminal emulation facility introduced enabling the use of a terminal on a local computer as a terminal on a remote computer thereby allowing control of file transfer sessions from a single terminal. *TSX PLUS version released for DEC PDP-11.*

1983: Option to control file transfers from command files as an alternative to control from operator's keyboard. *P/OS version released for DEC Professional.*

1984: Mechanism built into the package protecting against "message bouncing" due to line noise when computers remain connected and the package is not in use. *MicroRSX and MicroRSTS versions released for DEC MicroPDP-11.*

1985: Commenced development of new portable versions written in the programme language 'C'.

1986: First releases of new portable versions written in 'C' for PC-DOS, MS-DOS, UNIX, AIX and VMS.

1987: Portable versions support simultaneous multiple links. IPL-11 wins ICP Million Dollar Award.

1988: PC versions enhanced with improved terminal emulation including VT100 emulation, keyboard mapping and facilities to define function keys.

1989: Comprehensive upgrade for unattended operation of multiple PC/host links supporting auto-dialling modems.

1990: Release of MULTI-POLL range of polling software incorporating IPL-11 allowing PCs or other computers at multiple sites to be telephone-polled by a central host computer to transfer data to and from the central system.

All trademarks acknowledged.

XOREN COMPUTING LTD
28 MADDOX STREET
LONDON W1R 9PF
TELEPHONE 071-629 5932
FAX 071-629 5432

CIRCLE NO. 151

I am interested in Xoren Comms Software:

Name ____________________

Position ____________________

Company ____________________

Address ____________________

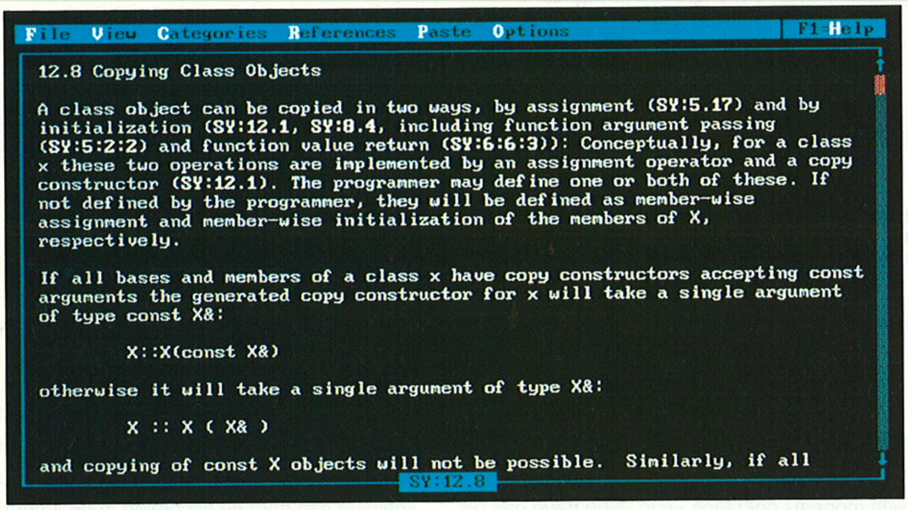
____

Postcode ____________________

EXE8/90

Return to: Xoren Computing, Freepost 27, London W1E 5JY.

Figure 3 - Screen shot of Help database.



final object code has the effect of 'distancing' a C++ translator vendor from the end product:- the executable program. Ultimately, it will be much more difficult for a company like Glockenspiel to provide such goodies as VROOM, VCM, and fully integrated debugging support than it will be for the likes of Borland, Zortech and other prospective manufacturers of native code compilers.

Until I've had a chance to check out the new Zortech release (which claims Windows 3 compatibility) I believe that Glockenspiel is still a reasonable choice for Windows and OS/2 development, although I do have some reservations about the lack of anything except large-model support. For regular MS-DOS work, I don't feel that Glockenspiel is a serious competitor to Turbo C++.

EXE

Dave Jewell is a Senior Consultant to the SignExpress Group, Basingstoke. He can be contacted on CIX as djewell, *where he helps moderate the* windows *conference.*

Glockenspiel C++ for the PC (together with CommonView) is available from QA Training (0285 655888) priced £495. Apart from the PC version, Glockenspiel's C++ is available on an enormous variety of platforms, including the recently introduced IBM RS/6000. Details of these are obtainable directly from the manufacturer in Eire (0001 733 166).

DON'T GIVE IT AWAY!

Software Developers! Protect Your Software Against Unlicensed Use.

From **Link Computer** Denmark, a remarkable piece of pure software that will lock a DOS .com or .exe file to a diskette, machine or network so that you may copy (or back-up) to your heart's delight, but you can still only run the program on the designated medium.

It's called **Cop's Copylock II** and it's clever stuff. You get an unlimited number of protections per purchase and its also uncrackable. Forget your favourite copying programs - they won't work on this. Forget Option Board and Bit Copiers - we dealt with them years ago.

If you don't want to dabble with dongles and aren't so keen on your software being freeware, you ought to take a look at this.

Ring Dino Cano on
Tel: 0763 263 073 Fax: 0763 262 620

Link Computer ApS.

CIRCLE NO. 152

CopyControl

THE NEW GENERATION OF COPY PROTECTION

NO User hassle
NO Back-up problems
NO Hardware add-ons or special disks needed
NO Changes to source code required
YES CopyControl beats ALL bit-copier Programmers
YES Floppy disks, hard disks and networks supported
YES CopyControl is totally transparent to the user
YES You can limit program use by no. of copies, no. of executions or date
YES CopyControl works on all IBM compatibles
YES Free demo disk available

For Further Information Phone or Write to:

microcosm
Microcosm Limited.
17 Cranbrook Road, Bristol BS6 7BL
Telephone: 0272-441230

CIRCLE NO. 153

POWER AND FLEXIBILITY

BRIEF 3.0 increases your productivity for just £159.00*

*VAT extra, FREE delivery

BRIEF has established itself as the leading programme editor on the market today. Now, with the launch of the latest enhanced version – BRIEF 3.0 – it offers even better productivity.

ENORMOUS UNDO POWER

BRIEF 3.0's enhanced UNDO capability takes the fear out of programming. BRIEF 3.0 allows you to undo any command that affects the cursor or the text (including global changes) up to 300 times, even back to the beginning of your session. It's the safety net you need to program with confidence.

REAL FLEXIBILITY

There's built-in support for 36 languages including Assembler, BASIC, C, COBOL, dBase, FORTRAN and Pascal. And thanks to its complete macro language you can make BRIEF 3.0 fit your own programming style with ease: change key assignment, commands, prompts, colours – even the indenting styles.

BRIEF 3.0 – new benefits

- With our new C-like macro language, CBRIEF, C programmers can write macros on the fly and not have to switch between C and a different macro language
- The new source level debugger works with both the original BRIEF syntax and the new CBRIEF macro language
- Maximum lines per file has been increased to 4 billion
- We've added to and improved template editing and smart indenting for ADA, Cobol, Basic, Fortran, Modula-2, Pascal and C
- We've added editable, multiple keystroke macros that can be saved and restored
- You can even save and restore your window set-ups

Plus the features that made BRIEF famous

- Flexible windowing
- Unlimited number of files
- File size limited only by desk space
- Regular expression search and replace

No other editor puts this much power and flexibility in your hands. So if you're looking for extra productivity purchase your copy of BRIEF 3.0 today.

Solution Systems

SOLUTION SYSTEMS are the UK distributors for BRIEF 3.0. When you buy from us you get the authorised UK version of BRIEF 3.0 dual media (3½" and 5¼") plus bonus macro diskette.

Special phone line for Access/Visa card holders. Save time by phoning 0763244141 between 9.00 and 6.00 Monday to Friday to placeyour order.

N.B Official purchase orders from local authorities, PLCs and educational establishments are accepted by post or by Fax.
Fax no: 0763 244025

BRIEF 3.0 The Programmer's Editor

Send to: Solution Systems UK, The Maltings, Green Drift, Royston, Herts SG8 5DB

Please rush me my copy of BRIEF 3.0

☐ **I enclose a cheque to the value of £182.85 (£159.00 + V.A.T.) crossed and made payable to Solution Systems UK.** **Orders despatched same day.**

NAME ______ ADDRESS ______

POSTCODE ______

☐ **Please charge my Access/Visa* account the sum of£182.85 (£159.00 + V.A.T.)**

Card No ______ **Expiry Date** ______ **Signature** ______

*Delete as appropriate

EXE 7/90

CIRCLE NO. 154

A Faster Way to Search Text

End 'waiting for lengthy string searches to finish' blues with the Boyer Moore algorithm! Jim Segrave introduces finesse to your text searches.

One often needs to search a text buffer for a string match. The most obvious approach - you might call it the Arnold Schwarzenegger algorithm - is to create an outer loop, which creeps through the buffer a character at a time, containing an inner loop, which compares each character of the test string with the current piece of buffer. This technique is demonstrated in the `brute_force` routine in Figure 1. The problem with it is that it must make at least one comparison for every possible starting location in the text buffer.

Boyer Moore searches make use of a very simple idea. The algorithm tries to decide if it's possible for the search pattern to occur at the current search point. If not, it skips forward to the next place in the text buffer where it might be possible.

How it works

To decide whether the search pattern might occur at the current search point, look at the character in the buffer which must match the end of the search pattern. One of three possibilities exist:

1) The character is the same as the last character of the search pattern. In this case, we will actually compare the characters beginning at search point to see if we've found a match.

2) The character is not one which is part of the search pattern - we can ignore all the characters through and including the one we just looked at, since its not possible that the search pattern can be found anywhere in the region from the search point through the character we've looked at. We move the search point forward by *search length* characters and try again, saving *search length* tests.

3) The character is not a match to the end of the search pattern, but it does occur somewhere in the search pattern. We can move the search point forward a bit, since we know that we can't find a match here. The amount we can move is determined by the position of the character we just found in the search pattern. Consider the situation with the search pattern set to `'YELLOW'`, and the character we pick up is a `'Y'`. We can move the search point forward to the `'Y'`, since any match to the pattern would have to begin at the `'Y'`. If the character appears more than once in the search pattern, we must be more careful. Suppose the character in the buffer is `'L'`. Figure 2 illustrates this situation. It is possible that `'YELLOW'` might occur at two places in the text buffer - the `'L'` might match either the first or second `'L'` in the pattern. We can only advance the search point by two characters, in case the `'L'` matches the second `'L'` of the pattern.

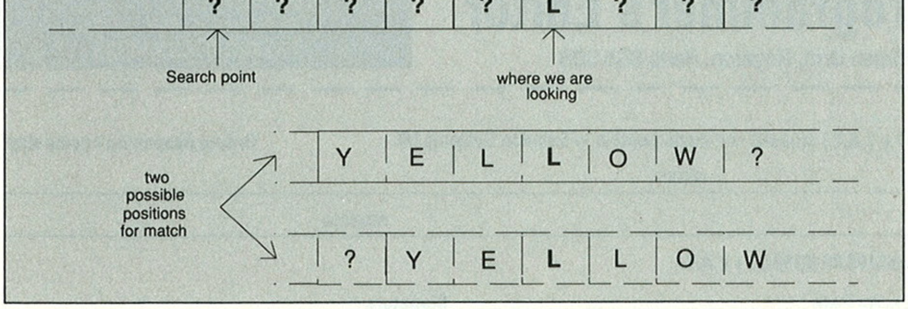


Figure 2 - Finding 'L' in the text buffer.

Figure 1 - brute_force ( ), the obvious approach.

```
char *brute_force (char *text,
                   int textlen,
                    char *search,
                    int searchlen)
  {
  char  *sp; /* current search point */
  char  *ep; /* last possible loc to try */
  int   i;

  ep = text + textlen - searchlen;
  sp = text;

  while (sp < ep)
    {
    /* the time eating inner loop */

    for (i = 0; i < searchlen; ++i)
      if (sp[i] != search[i])
        break;

    if (i == searchlen)
      return (sp);     /* found it!  */
    /* advance search point and loop  */
    ++sp;
    }

  return (NULL); /* not found...  */

  }
```

It is not very efficient to look for the character in the search pattern and calculate how far to advance the search point each time. It is far better to build a complete table of search point adjustments before we begin. For every possible character, the table will contain the number of characters to advance the search point if we find that particular character in the buffer when we look for a match for the end of the search pattern.

We initialise every entry in the table to the length of the search pattern - if a character does not appear in the search pattern, case 2 above, then we skip forward by the search length.

We then read through the search pattern, setting the table entry to *search length - offset of char in search pattern - 1.* These

★ POCKET SIZED ★ PC-COMPATIBLE ★ PERSONAL ORGANISER ★

ATARI Portfolio

THE PC IN YOUR POCKET

The new Portfolio from Atari is the world's first pocket-sized electronic organiser that is also a powerful PC compatible computer, with a full QWERTY keyboard and scrolling 80x25 character display.

Portfolio includes everything you would expect from an electronic organiser - an address and telephone book, time manager, diary and sophisticated calculator.

Portfolio also includes several functions that you would not expect - a spreadsheet for your personal budget and expense records, as well as a text processor for typing memos and letters.

And, because Atari's Portfolio is PC compatible, it can communicate with your desktop PC at home, or in the office. You can transfer files from one to the other, to enable you to update your reports and figures with the Portfolio while you are on the move.

FOR ONLY **£249** INC VAT

SPECIFICATION

- ★ ***Processor:*** Intel 80C88 at 4.9152Mhz.
- ★ ***Operating System:*** Compatible with MS-DOS 2.11.
- ★ ***Internal ROM:*** 256K containing BIOS, operating system and applications suite.
- ★ ***RAM:*** 128K with an internal RAM disk, configurable from 8K. Externally expandable to 640K RAM.
- ★ ***Keyboard:*** 63 keys, QWERTY, IBM PC BIOS compatible. Buried numeric pad and function keys. Optional key click.
- ★ ***Character Set:*** Extended IBM ASCII (255 characters).
- ★ ***Mass storage:*** credit card sized memory cards (32K or 64K or 128K RAM).
- ★ ***Display:*** Graphics LCD, supertwist technology, MDA compatible, 40 columns x 8 lines, 240 x 64 pixels (with the option to window a full 80 x 25 character display). Keyboard controlled contrast.
- ★ ***Peripherals:*** 60 pin expansion BUS to take serial and parallel ports and memory expansion units.
- ★ ***Size:*** 8" x 4" x 1" (200mm x 105mm x 29mm).
- ★ ***Weight:*** 495 grammes (with batteries).
- ★ ***Applications:*** calendar and diary, address and phone book, Lotus 1-2-3 compatible spreadsheet, text processor, communications software.

COMING SOON!

In addition to the excellent software built-in to the Portfolio Free of Charge (see right), other software and peripheral products, such as the sophisticated Pocket Finance package and serial/centronics interfaces are available. And it doesn't stop there. Many manufacturers have recognised the potential of the Portfolio and have already started to design new peripherals and software. Products currently under development include: Serial interface with built in mini modem, Apple Macintosh interface, business, utility and programming software plus a range of adventure and battle strategy games. For further free details on the Portfolio range, fill in the coupon below and return it to Silica Systems now.

5 BUILT-IN APPLICATIONS TO GET YOU ORGANISED!

To get you started, Portfolio comes with a suite of five useful functions built-in, all accessible from a simple menu display.

TIME MANAGER

More than just an electronic diary, the Portfolio Time Manager enables you to plan your appointments via a comprehensive calendar and diary. It can even be programmed to sound an audible alarm at specific times to remind you of important appointments.

CALCULATOR

Your Portfolio will be invaluable in the office or at home as a powerful pocket calculator. It has a full range of functions, including factorial, power and root calculations, all with multi display formats and memories.

ADDRESS BOOK & DIALLER

Portfolio has a complete address book facility that allows you to store hundreds of addresses and phone numbers. And, at the touch of a button, you can retrieve any one of them, or search for a specific grouping, such as "all Italian restaurants". And, when you are ready to book your table, hold your Portfolio to your telephone mouthpiece and use its special built-in tone dialler to dial the number for you.

TEXT PROCESSOR

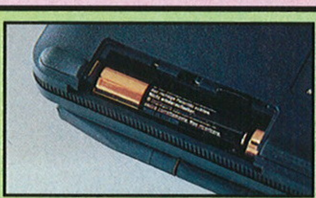
The Portfolio's built-in text processor program includes word wrap, line and column count, string search, in fact most of the functions you would find in a word processor. It handles printer and word processor control codes and allows easy transfer of files between Portfolio and your desktop PC.

SPREADSHEET

For real calculating power, Portfolio has a Lotus 1-2-3 compatible spreadsheet built-in. It has 127 columns x 225 rows and reads/writes Lotus V1.0 and V2.01 files, so you can transfer data to and from Lotus 1-2-3 on your desktop PC. The Portfolio's 256K ROM includes MS-DOS and PC BIOS compatible systems software.

MEMORY CARDS

Portfolio can store and retrieve data and programs from its own RAM, or from small credit card size memory cards, that slot into its built-in card drive. The cards are available in three sizes, 32K, 64K and 128K, so you can carry a library of data in your pocket. The card drive also accepts ROM cards, which can contain commercial or custom software.



POWER SUPPLY

Portfolio is powered by three AA batteries which will run for up to six weeks with normal use, or from the mains using an adaptor. All the peripherals take their power from the Portfolio, so no extra batteries or adaptors are required. A 'battery-low' warning and memory back-up ensure that information is not lost when the batteries are changed.

INTERFACES & PERIPHERALS

Portfolio can communicate with other computers and supports a growing range of peripherals via a built-in 60 pin bus connector. Peripherals available include serial and parallel interfaces and memory expanders (to 640K). You can also add a card drive to your desktop PC, to enable it to read/write to Portfolio's cards.

SILICA SYSTEMS OFFER YOU

FREE OVERNIGHT COURIER DELIVERY: On all hardware orders shipped in the UK.
TECHNICAL SUPPORT HELPLINE: Team of Atari technical experts at your service.
PRICE MATCH: We normally match competitors on a "Same product - Same price" basis.
ESTABLISHED 12 YEARS: Proven track record in professional computer sales.
£13M TURNOVER (with 60 staff): Solid and reliable with maintained growth.
BUSINESS/EDUCATION/GOVERNMENT: Volume discounts available for large orders.
SHOWROOMS: Demonstration and training facilities at our London & Sidcup branches.
THE FULL STOCK RANGE: All of your Atari PC requirements from one supplier.
FREE CATALOGUES: Will be mailed to you with offers and software/peripheral details.
PAYMENT: By cash, cheque and all major credit cards.
CREDIT PAYMENT TERMS: Silica are licensed credit brokers - details on request.

Before you decide when to buy your new Atari Portfolio, we suggest you think very carefully about WHERE you buy it. Consider what it will be like a few months after buying your Portfolio, when you may require additional peripherals or software, or help and advice with your new purchase. And, will the company you buy from contact you with details of new Portfolio products? At Silica Systems, we ensure that you will have nothing to worry about. Silica Systems is a new division of Silica Shop, the UK's leading Atari specialists. This new division has been established to provide a service to the more serious home user, as well as to business and education purchasers. Silica have been established for over 12 years, and have an annual turnover of £13 million. With our unrivalled experience and expertise, we can now claim to meet our customers requirements with an understanding which is second to none. But don't just take our word for it. Complete and return the coupon now, for our latest Free literature and begin to experience the "Silica Systems Service".

SILICA SYSTEMS

MAIL ORDER: Order Lines Open: Mon-Sat 9.00am-6.00pm	1-4 The Mews, Hatherley Rd, Sidcup, Kent, DA14 4DX	No Late Night Opening	**Tel: 081-309 1111** Fax No: 081-308 0608
LONDON SHOP: Opening Hours: Mon-Sat 9.30am-6.00pm	52 Tottenham Court Road, London, W1P 0BA	Late Night: Thursday until 8pm	**Tel: 071-580 4000** Fax No: 071-323 4737
SIDCUP SHOP: Opening Hours: Mon-Sat 9.00am-5.30pm	1-4 The Mews, Hatherley Rd, Sidcup, Kent, DA14 4DX	Late Night: Friday until 7pm	**Tel: 081-302 8811** Fax No: 081-309 0017
BUSINESS/EDUCATION: Order Lines Open: Mon-Fri 9.00am-6.00pm	1-4 The Mews, Hatherley Rd, Sidcup, Kent, DA14 4DX	Closed on Saturdays	**Tel: 081-308 0888** Fax No: 081-308 0608

To: Silica Systems, Dept EXE-0690-29, 1-4 The Mews, Hatherley Rd, Sidcup, Kent, DA14 4DX

PLEASE SEND INFORMATION ON PORTFOLIO

Mr/Mrs/Ms: .... Initials: .... Surname: ....

Address: ....

....

EXE 8•90

.... Postcode: ....

Tel (Home): .... Tel (Work): ....

Company Name: ....

Which computer(s), if any, do you own? ....

E&OE - Advertised prices and specifications may change - please return the coupon for the latest information.

CIRCLE NO. 155

are the values we use for case 3 above - where we find a character which isn't a match for the end of the search pattern but does occur somewhere in it. This scheme automatically handles the problem of multiple occurrences of a character in the search pattern, since the value will be rewritten as we go through the search pattern. The new value will be smaller for the second (or third or ...) occurrence of the character in the search pattern.

For the search on 'YELLOW' above, this means that `table['Y']` would become 5 (= *search length - 1*), `table['E']` would be 4, `table['L']` would be set first to 3 and then to 2, and `table['O']` would be set to 1. All remaining characters take the value 6.

The final character of the search pattern wants special handling. If we just use the method in the previous paragraphs, we would write a zero in the table for this character. That's fine, the zero tells us that it is worth actually examining the buffer at the search point for the search pattern, but it doesn't tell us what to do if we don't find the pattern there. We need save the current table value before setting it to zero. We'll use this saved value to decide where to go when, after finding the last character of the pattern in the right place, we discover that we haven't got a match.

For example: if the search pattern is `'BELLE'`, when we find an `'E'` at search point +4, `table['E']` will be zero and we go off and check for `'BELL'`. If we don't find it, we should move the search point forward 3 characters, since the `'E'` we were looking at could be a match to the second character of the search pattern rather than the last.

```
char *boyer_moore (char *text, int textlen, char *search, int searchlen)
  {
  int   table[256];      /* assuming 8 bit characters    */
  int   lastjmp;       /* how far to go if a test fails  */
  int   i;
  unsigned char  *sp;    /* current search point       */
  unsigned char  *ep;    /* last possible loc to try    */
  unsigned char  c;

  /* The use of unsigned chars is a necessity, since we are */
  /* going to use the char values as array indices       */

  /* set up the table of adjustments to the search point    */
  /* initialise for any char which isn't in search       */

  for (i = 0; i < 256; ++i)
    table[i] = searchlen;

  /* now reset table for chars in search pattern        */

  for (i = 0; i < searchlen -1; ++i)
    {
    c = (unsigned char) search[i];
    table[c] = searchlen - i - 1;
    }

  /* save value for last char and mark it as special      */

  c = (unsigned char) search[searchlen - 1];
  lastjmp = table[c];
  table[c] = 0;

  /* the actual search routine  */

  ep = (unsigned char *) text + textlen - searchlen;
  sp = (unsigned char *) text;

  while (sp < ep)
    {
    /* c = char which should match the last char search pattern */

    c = sp[searchlen -1];

    if (table[c] != 0)
      {
      /* not a match for last char, adjust the search  */
      /* point using the table and try again       */

      sp += table[c];
      continue;
      }

    /* This could be a match, at least the last char is right. */
    /* Actually compare strings at the search point.      */

    for (i = 0; i < searchlen; ++i)
      if (sp[i] != search[i])
        break;

    if (i == searchlen)
      return ((char *) sp);  /* found it!  */

    /* False alarm - it wasn't a match. Now we need the  */
    /* table value for the last char of the search pattern */
    /* which we saved for just this purpose...        */

    sp += lastjmp;

    }

  return (NULL);          /* buffer exhausted  */

  }
```

Figure 3 - Implementation of Boyer Moore.

Once this array is set up, we will be able to skip from 1 to *search length* characters for every character we actually examine in the buffer. For all but a few pathological cases, this will save an enormous number of iterations of the inner search loop. An example where it will not help much would be a search pattern of `'ab'` in a text buffer of all `'a'`s. It is also ill-suited to searches of very small text buffers, where the overhead of setting up the table is significant compared to the time for brute force searching.

The algorithm is guaranteed to terminate for all non-empty search patterns, since every entry in the table except the one for the last character of the search pattern will contain some positive value, as will the saved adjustment for the last character of the pattern. Therefore, every time we look in the buffer, we will either find the pattern or advance the search point, so sooner or later we will have examined the entire buffer.

Figure 3 is a straightforward implementation of the above. The code is far from optimised - it can be tightened up, but it will still be faster than the tightest assembler version of the brute force method.

Testing it out

The crude test harness shown in Figure 4 (run on a .DOC file of medium size, searching for a text string which occurs just once at offset about 31000 in the file) gave timings of 27.69 seconds for 100 brute force searches, 3.68 seconds for 100 Boyer Moore searches on a 16MHz 386SX.

Here are some statistics from this test -

Brute force - 31423 outer loops, 31557 inner loops. The inner string compare bombs out on the first character almost every time, but it executes often - there's little profit in making the compare faster, since it almost always terminates quickly. Any real savings to be made will be in setting up for the inner loop and testing for termination of the outer loop.

Boyer Moore - 3177 outer loops - we only ever looked at about one character in ten! 856 inner loops - of these 3177 looks, only 856 were occurrences which matched on the last character.

The search pattern used ends in a space - a very common character in a .DOC file. I deleted the space and repeated the test with the shorter string. The timings were virtually unchanged (down to 3.46 seconds from 3.68), but the statistics became

Brute force - 31423 outer loops, 31556 inner loops.

ASPECT

The definitive database for micros and micro-networks.

MAIN FEATURES

- Clear Concise English-like programming language.
- Rapid system development and easy maintenance.
- Relational and Hierarchical structures.
- Variable length records.
- Powerful facilities for reading and writing files from other systems - e.g. Pegasus, Lotus 123.
- Run-time licences available.

OS/2, PC-DOS, MS-DOS, XENIX,
CDOS, MS-NET, NOVELL ETC.

Microft Technology Limited
The Old Powerhouse, Kew Gardens Station
Kew, Surrey TW9 3PS.
Telephone: 081-948 8255

£560+VAT Single User.
£950+VAT Network Version up to 5 Users.
£100+VAT each subsequent User.

CIRCLE NO. 156

Figure 4 - Test Harness to compare two search techniques.

```
#include  <stdio.h>
#include  <stdlib.h>
#include  <string.h>
#include  <bios.h>

static char buf[32000];
static char *search = "14H dispatcher ";   /* known to be in file */

void main (void)
  {
  FILE  *fp;
  int    i;
  long  start, stop, delta;

  /* an arbitrary choice of a text file to test on  */

  if ((fp = fopen ("d:/temp/mnpdrv/fossil.doc", "r")) == NULL)
    {
    perror ("Can't open test file");
    exit (1);
    }

  if (fread (buf, 32000, 1, fp) != 1)
    {
    perror ("Can't read test file");
    exit (1);
    }

  start = biostime (0, 0L);        /* Turbo C timer function */

  for (i = 0; i < 10; ++i)
    {
    if (brute_force (buf, 32000, search, strlen (search)) == NULL)
      {
      printf ("Brute force: not found\n");
      break;
      }
    }

  stop = biostime (0, 0L);
  delta = (stop - start) * 1000L;   /* 1000 * number of ticks    */
  delta /= 182;                     /* at 18.2 ticks/sec - gives  */
                                    /* seconds * 100             */
  printf ("Brute force: %ld.%02ld\n", delta / 100L, delta % 100L);

  start = biostime (0, 0L);

  for (i = 0; i < 10; ++i)
    {
    if (boyer_moore (buf, 32000, search, strlen (search)) == NULL)
      {
      printf ("Boyer Moore: not found\n");
      break;
      }
    }

  stop = biostime (0, 0L);
  delta = (stop - start) * 1000L;
  delta /= 182;

  printf ("Boyer Moore: %ld.%02ld\n", delta / 100L, delta % 100L);

  exit (0);

  }
```

Boyer Moore - 3222 outer loops, 158 inner loops - we've reduced the number of times we bothered with any further testing by a factor of 5. The fact that this did little in terms of performance suggests that possibly the bulk of the time is going in the outermost loop or the setup of the table itself.

The test code was compiled under Turbo C V2.0, small model. The test harnesss makes use of the Turbo library function `biostime();` but the rest of the code is in generic C. Last one to write a hot `grep` is a sissy!

EXE

JimSegrave is a freelance contract programmer specialising in system software. He can be reached as jes on CIX.

PURE MAGIC★

COMPUTER AIDED PROGRAMMING
APPLICATION GENERATOR &
RELATIONAL DATA BASE

BEYOND 4 GL!

— THE GOALS —

★ Well-engineered applications for end-users
★ Up to 90% reduction in development effort
★ Minimal on-going maintenance costs

— THE SOLUTION —

MAGIC
Computer Aided Programming
Application Generator and Relational Database

— THE MEANS —

★ Fully integrated Database and Application ENGINE
★ **For developers,** the ENGINE drives a special application - The Computer Aided Programming Application - eliminating the need for any 4GL.
★ **For end-users,** the ENGINE drives the developed application with user-friendly facilities conveniently specified by the developer through easy-to-use tables

— THE PROOF —

MARCH '89, AUSTRALIA: ***MAGIC*** **- Outright winner in PC WEEK Database Challenge**
JANUARY '89, USA: ***MAGIC*** **- Shared Top Honours in PC WEEK Poll of Corporate Satisfaction**

"There is no competitive product"
(PC TECH JOURNAL)
"... this program is a stellar performer"
(PC WORLD)
"One of the most powerful DBMS packages available" (PC MAGAZINE)

— THE ULTIMATE PROOF —

Thousands of experienced, knowledgeable, professional developers in over 20 countries have switched from better-known 'brand-name' products to ***MAGIC***

— THE ENVIRONMENTS —

OPERATING SYSTEM: PC-DOS/MS-DOS 2.0 and up
HARDWARE: IBM-PC/XT/AT, IBM PS/2 and full compatibles, 640K RAM
NETWORK SUPPORT: Any DOS 3.1 compatible network including Novell Netware and IBM Token-Ring
Versions also available for Unisys-BTOS, Convergent-CTOS.

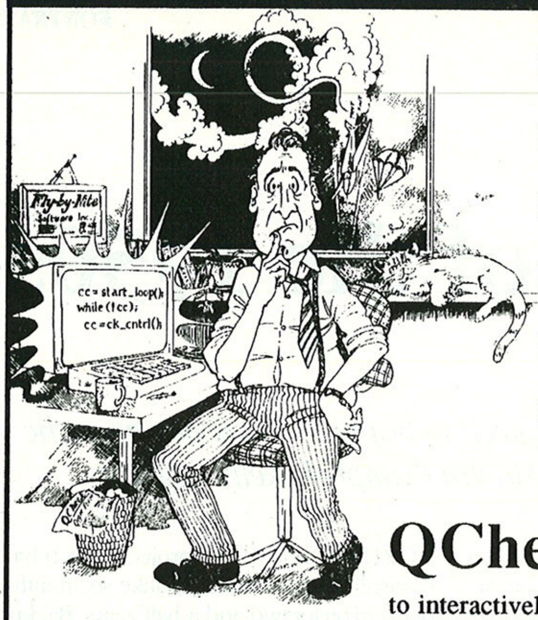
MSE

UK ENQUIRIES:
MSE (UK) LTD, Fulton House, Fulton Road, Empire Way, Wembley, Middx HA9 0TF
TEL: 01-902 8998 FAX: 01-903 1465

All trademarks and copyrights are acknowledged

CIRCLE NO. 157

THE SOFTWARE TESTING NIGHTMARE ?



- Are you serious about the way you develop programs?
  (Of course I am)
- Do you test your programs?
  (Doesn't everybody?)
- How much do you test your programs?
  (I try out all, well most, of its functions)
- Do you look forward to testing your programs?
  (Who does?)
- Can you quantify how much you test your programs?
  (How could I, even if I wanted to?)
- Have you considered the implications of not being able to demonstrate that you have fully tested your program?
  (...Ah, um...not really yet)

QCheck is a unique program that enables professional software developers to interactively find out how much their program has been exercised. It also provides objective evidence in the form of listings showing unexecuted lines. For DOS high level language compilers. A single user licence is **£150.00**+VAT *(how much does **just one** undetected bug cost you?)*

Also - **ETP** an Epson™ to Postscript™ translation utilty that allows existing applications to print to your Postscript laser printer, whether connected locally or on your network. A single user licence costs **£55.00** + VAT.

Telephone 0883-341697, Fax 0883-341343

Seltek Ltd., Seltek House, 38 Westway, CATERHAM, Surrey. CR3 5TP.

SELTEK

CIRCLE NO. 158

C-Compilers ONLY £595

Developed with 12 years experience, true processor specific, ANSI C-Compilers are now available complete with free Assembler and Linker.

1802, 1805, 6301, 64180, 6502, 6800, 6801, 6802, 6805, 6809, 6811, 68000, 8051, 8085, 8086, 8096, Z80......

Development Systems with High Level Debugging are also available to see you through your complete software and hardware design cycle.

Call us now for your **FREE** active demonstration disk.

american automation
uk oxford

Tel: (0993) 778991 Fax: (0993) 702284

Lea View House, Two Rivers Est, Station Lane, Witney, Oxon OX8 6BH

CIRCLE NO. 176

Middlesex Polytechnic

VENTURA PUBLISHER

Intensive one day courses introducing the participant to the world of desk top publishing using one of the leading **DTP** packages - **VENTURA.**

Each course member will be allocated an individual workstation together with fully documented course notes. Highly skilled trainers will be on hand to solve each individuals problems.

The hands-on course will enable the participant to design their own document layouts to suit their individual needs, the course will cover:

- Integration of **DTP** with well known software packages
- Editing text within a publication
- Text and paragraph layout - *creating your own styles*
- Use of frames in a document - *sizing and scaling*
- Placing pictures in a document - *anchors,captions and figure numbering*
- Using headers and footers - *page numbering*
- Automatic indexing and generation of contents lists
- Printing chapters and publications

Course Dates: 11th, 13th, 18th, 20th September 1990

Cost: (Including Lunch) £250 per person +15% VAT

Contact: Christine Watson on **081-368-1299 x7453**

CIRCLE NO. 177

FORTRAN: A Child of the 1950s

Some people think FORTRAN should never have been allowed to happen - yet it has been the greatest programming productivity lever of all time. Martin Campbell-Kelly explains.

FORTRAN gets a terrible press. Edsger Dijstra once said it 'is hopelessly inadequate for whatever computer application you have in mind today... too clumsy, too risky and too expensive'. FORTRAN has been blamed for the unstructured programming habits of a generation of programmers. And more spectacularly, FORTRAN was singled out as the underlying reason that the Mariner spacecraft - intended to be the first United States spacecraft to visit another planet (Venus) - had to be destroyed four minutes after lift-off on 22nd April 1962.

Yet the fact remains that for over 30 years, FORTRAN has been the most popular language for serious engineering and scientific applications. In number-crunching environments such as the aerospace industry, finite-element analysis, meteorological forecasting and high-energy physics, FORTRAN is almost universal. Often this is attributed to the triumph of IBM's marketing skill over its customers' judgement. But the real reason is encapsulated in the well-known epigram: *FORTRAN is a language to avoid - unless you want some answers.* The essential characteristics of FORTRAN are its efficiency, popularity and standardisation. It became popular because it was efficient; it became a standard because it was popular; and once it became a standard, FORTRAN was here for good.

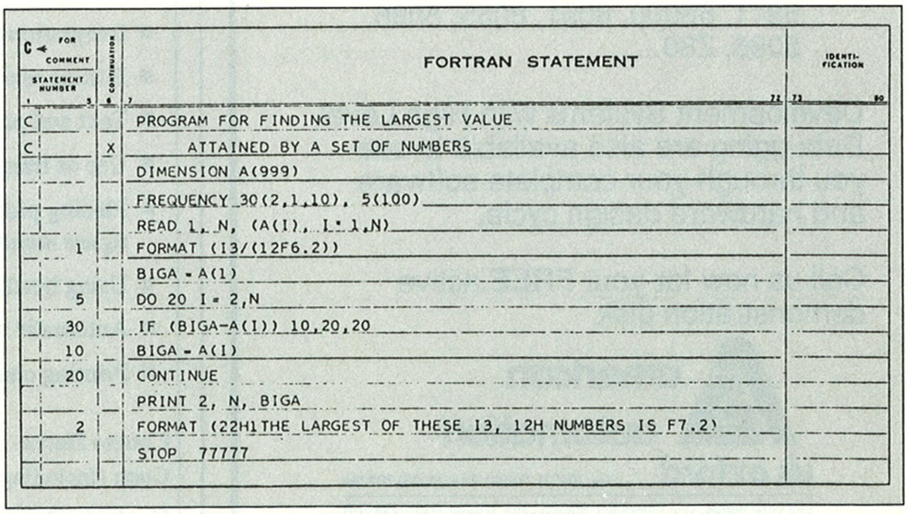
The origins

FORTRAN was invented in 1954. At that time, there were less than 100 computers world-wide, and the machine-code programmer was king. There had certainly been some attempts at 'automatic coding' - that is, getting a computer to do its own programming - but they had achieved very limited success. Compilations tended to be very slow (taking as much as half-an-hour on a $1 million machine) and the resulting programs were very inferior to those produced by a good machine-code programmer. The few automatic-coding systems that had been popular had mainly succeeded because their manifest inefficiency could be hidden behind the overhead of software floating-point routines. But the arrival of the IBM 704 computer with hardware floating point meant this would no longer be possible: there would be no place to hide object-code inefficiency. Whether it was possible to produce automatically machine code as good as that produced by a human programmer was a high-running controversy at the time; and a lot of mathematicians and engineers had staked their careers as machine-code programmers on the assumption it was not.

John Backus, an IBM high-flyer in his late twenties, had been the co-designer of the new model 704 with Gene Amdahl. In late 1953, he successfully proposed that he should organise a Programming Research Group to develop an automatic-coding system that would produce programs that were at least 90% as efficient as those written by a machine-code programmer. This was a major technical challenge, and not many people believed it could be done - including plenty of people in IBM. According to one of his contemporaries, there was a feeling that Backus had 'bitten off more than he could chew' and one IBM manager called it the 'FORTRAN fiasco'. The project, which had originally been intended to take six months, dragged on for two-and-a-half years. Backus himself was not immune to the criticism from the doubters and came close to giving up his leadership of the project; but he went on to complete one of the most remarkable pieces of programming in computer history. The compiler contained 18,000 machine instructions and took about 25 programmer-years of effort. The final cost was estimated by IBM at $475,000.

The language

FORTRAN was first defined in an historic document *Specifications for the IBM FORmula TRANslating System,* dated 10th November 1954. The primary goal of the FORTRAN project was object program efficiency, so that very little thought was given to the design of the language itself - as Backus later recalled, 'we simply made up the language as we went along'. One regrets this now, of course, but the idea of creating in FORTRAN a standard programming language for the future was quite inconceivable at the time. FORTRAN was designed for object program efficiency: designing an optimising compiler that would generate excellent



```
C ← FOR COMMENT | STATEMENT NUMBER | CONTINUATION | FORTRAN STATEMENT | IDENTIFICATION
C         PROGRAM FOR FINDING THE LARGEST VALUE
C        X     ATTAINED BY A SET OF NUMBERS
          DIMENSION A(999)
          FREQUENCY 30(2,1,10), 5(100)
          READ 1, N, (A(I), I = 1,N)
    1     FORMAT (I3/(12F6.2))
          BIGA = A(1)
    5     DO 20 I = 2,N
   30     IF (BIGA-A(I)) 10,20,20
   10     BIGA = A(I)
   20     CONTINUE
          PRINT 2, N, BIGA
    2     FORMAT (22H1THE LARGEST OF THESE I3, 12H NUMBERS IS F7.2)
          STOP  77777
```

Figure 1 - 'A complete but simple program'

CALL FOR OUR SUMMER CATALOGUE

ASSEMBLERS
386 ASM Link & Locate	£550.00
MASM v.5.1	£89.00
OPTASM	£85.00
Turbo Assembler/Debugger	£89.00

BASIC
Microsoft Professional BASIC	£270.00
BASIC Interpreter	£215.00
Quick Basic *U.K*	£60.00
True Basic	£65.00
Power Basic	£65.00

C
Lattice C Compiler v.6.0	£170.00
Microsoft C v.6.0 *NEW*	£250.00
Watcom C v.8.0 *NEW*	£255.00
C-terp	£195.00
C86 Plus	£325.00
Express C	£65.00
High C-286	£385.00
High C-386	£585.00
Instant C	£310.00
Instant C/16M	£515.00
NDP C	£385.00
Quick C *U.K*	£60.00
Turbo C v.2.0	£65.00
Zortech C Compiler	£45.00

C++
Advantage C++	£310.00
Borland C++	£129.00
Borland C++ Professional	£195.00
Guidelines C++ with book	£275.00
Intok C++ 80386	£320.00
Zortech C++ Compiler v.2.1	£120.00
Zortech C++ Dev. Edition	£270.00

COBOL
acucobol MS-DOS	£900.00
acucobol Net-Bios	£1250.00
acucobol XENIX-286	£1500.00
acucobol UNIX-386	£2000.00
acucobol *OTHERS*	CALL
Microsoft COBOL	£585.00
Real Menu	£210.00
RealCICS	£750.00
Realia COBOL	£750.00
RM/Cobol-85	£750.00

dBASE
Clipper Version 5.0 U.K. *New*	£325.00
DB Fast/PLUS	£255.00
DB Fast/WINDOWS	£255.00
dBASE III Plus	£395.00
dBASE IV	£515.00
dBMAN V	£190.00
dBXL Diamond	£165.00
dBXL/Lan	£389.00
Foxbase+ v.2.10	£255.00
Foxbase+/386	£385.00
Foxbase+/LAN	£385.00
Foxbase+ Runtime	£445.00
Foxbase+ SCO XENIX	£495.00
Quicksilver Diamond	£389.00

DATABASE MANAGEMENT
Advanced Revelation	£615.00
AskSam	£190.00
Clarion Personal Developer	£195.00
Clarion Professional	£515.00
Opus One	£255.00
Oracle for 1-2-3	£195.00
Paradox 2.0	£469.00
PFS Professional File	£165.00
R:BASE Ext. Report Writer	£190.00
R:BASE for OS/2	£579.00
R:BASE for DOS	£445.00
Reflex	£89.00

FORTRAN
Microsoft Fortran v.5.0 *U.K*	£250.00
Fortran-F77L	£385.00
NDP FORTRAN-386	£385.00
ProFortran	£235.00
RM/Fortran	£385.00
SVS Fortran	£435.00

OTHER LANGUAGES
ACTOR	£450.00
ACTOR Language Extension	£65.00
CCSM Mumps Single User	£65.00
PC Logo	£65.00
RPG II Complete	£1035.00
Smalltalk/V	£65.00
Smalltalk/V 286	£129.00
Snobol 4+	£65.00
Trilogy	£65.00

PASCAL
Microsoft Pascal	£170.00
Oregon Pascal-2	£149.00
ProPascal	£235.00
Metaware Professional Pascal	£579.00
Metaware Prof. Pascal 286	£385.00
Turbo Pascal v.5.5 with objects	£89.00
Turbo Pascal Professional	£159.00

PROLOG
Arity Combination Pack	£710.00
Expert System Dev. pack	£190.00
Intelligence Compiler	£299.00
MicroPROLOG Professional	£645.00
MProlog	£299.00
Professional Prolog II Plus	£899.00
Professional Prolog	£545.00
Prolog Advanced Toolkit	£99.00
Prolog Compiler	£129.00
Prolog Compiler/Interpreter	£420.00
Prolog-86 Plus	£165.00
SQL Development Pack	£190.00
Turbo Prolog	£65.00

AI AND OBJECT ORIENTED
C-Talk	£99.00
C+ Objects Vol 1 with Source	£275.00
CxPERT	£515.00

COMMUNICATIONS
AdComm	£190.00
C ASYNCH MANAGER	£125.00
Communications Library	£195.00
Comm Pak with Breakout	£195.00
Multi Comm	£89.00
Net Lib	£129.00
NET-TOOLS	£95.00
SilverComm	£125.00
Comm Plus	£65.00
Turbo ASYNCH PLUS	£125.00

FILE MANAGEMENT
asmTREE	£245.00
B-Tree	£65.00
B+Tree for C	£55.00
AXOS	£255.00
Btrieve v.5.01 Novell	£165.00
Btrieve OS/2	£370.00
Btrieve Multitasking	£370.00
Btrieve/N v.5.01	£395.00
Btrieve XENIX	£370.00
C Index+	£2545.00
c-tree	£255.00
Faircom Toolbox Professional	£710.00
Code Base 4	£190.00
db_FILE/RETRIEVE	£190.00
d-tree	£320.00
dBC III	£165.00
dBC III Plus	£330.00
RealDL/I	£645.00
WKS Library	£129.00

GENERAL
ASMLIB	£89.00
C Tools Plus	£99.00
C Utility	£165.00
C*Lib C Function Library	£120.00
DB/RA3	£125.00
dBase Program Utility	£55.00
dBase Program Utility Vol 2	£55.00
dBase Tools For C	£55.00
Finally	£60.00
Functions with source	£149.00
Integrated Dev. Libraries	£99.00
Pascal Toolkit	£85.00
Pascal Tools 1	£75.00
QuickPak Professional	£99.00
Quick Tools Library	£79.00
Spindrift Library	£99.00
Super Functions	£195.00
Tom Rettig Help Combo	£79.00
Tom Rettig Library	£65.00
Turbo C TOOLS	£99.00
Turbo Plus	£129.00
Turbo PowerTools Plus	£99.00
Turbo Functions	£69.00

GRAPHICS
dBase Tools for C Graphic	£55.00
dGE v.3.0	£125.00
EGA Toolkit	£165.00
Essential Graphics	£259.00
Flipper	£129.00
FontWINDOW/Plus	£170.00
GEOGRAPH Professional	£210.00
Grafmatic CGA & EGA	£135.00
GraphiC	£255.00
GSS Graphics Dev. Toolkit	£399.00
GSS Kernel	£385.00
GSS Metafile Interpreter	£225.00
HALO-88	£255.00
HOOPS Extended DOS	£515.00
MetaWINDOW/Plus	£210.00
scLASERplus	£299.00
SilverPaint	£65.00
SilverPak	£179.00
Slate	£195.00
Universal Graphics Library	£130.00

OP. SYSTEM ENHANCEMENTS
C Sharp	£300.00
DESQview API C Lib/Ref	£125.00
Multi C	£89.00
Multi Dos Plus	£60.00
OS/2 Programmer's Toolkit	£215.00
OS/286	£299.00
OS/386	£299.00
Resident C	£120.00
Sapiens V8	£180.00
Synergy Toolkit	£239.00
Time Slicer	£179.00
VMFM	£115.00
Wendin PC UNIX	£60.00
Wendin PC VMS	£60.00
Windows Development Kit	£295.00

QUERY
CBTREE/SQL	£239.00
CQL Query System	£239.00
dQUERY/Lib	£179.00
db_RETRIEVE	£239.00
XQL	£480.00
Xtrieve PLUS	£360.00
Netware SQL	£385.00

REPORT
CB/IQ Report	£139.00
Coreports	£179.00
Focus Report Writer	£210.00
r-tree	£179.00
R&R Relational Reportwriter	£95.00
R&R Clipper Module	£49.00

TEXT SCREENS
Animator COBOL	£120.00
Application Manager	£450.00
C-Worthy v.1.1	£160.00
C-Worthy + Forms	£240.00
C-Worthy + Forms + Source	£395.00
C-Worthy All C compilers	£480.00
C-Worthy Forms only	£80.00
C-Worthy Source code only	£155.00
C-Worthy OS/2	£240.00
C-Worthy + Forms for OS/2	£320.00
C-Worthy + Forms + Source	£480.00
C-Worthy OS/2 & MS-DOS	£315.00
C-Worthy + source	£795.00
Curses	£150.00
Curses Window Dev. Kit	£75.00
Data & Windows	£149.00
DataWindow	£179.00
dWindows for db3	£60.00
Face-it	£60.00
Flash-up w/Toolbox	£85.00
Forms II	£179.00
HI-SCREEN XL	£89.00
Multi Windows	£179.00
On-Line Help	£90.00
PANEL PLUS	£299.00
PC Forms	£90.00
Pop Screen	£55.00
POWER SCREEN	£79.00
Pro Screen	£60.00
SCREENIO	£245.00
Screen Sculptor	£75.00
ScreenStar	£120.00
Screenplay	£110.00
SoftCode	£120.00
SoftCode Template	£30.00
Softscreen/Help Source	£150.00
Sourcewriter	£605.00
UI Programmer Dev. Edition	£345.00
V Screen	£39.00
VCScreen	£90.00
Vitamin C	£135.00
VLib with Source	£90.00
Vermont Views	£245.00
Z View	£149.00

DEBUGGERS & DISASSEMBLERS
386 Debug	£125.00
Breakout-II	£79.00
D'Scope 2	£159.00
dBug for dBase	£125.00
DIS*DOC	£79.00
Periscope II	£110.00
Periscope II-X	£90.00
Periscope III	£875.00
Periscope I Rev 3	£435.00
Periscope IV (16 MHz)	£1245.00
Periscope IV (20 MHz)	£1435.00
Periscope IV (25 MHz)	£1625.00
Plus Board with 512K	£310.00
Plus Board with 0K	£220.00
Serial Test	£310.00
Soft Probe II/TX	£245.00
Soft-X-plore	£79.00
Sourcer + BIOS Pre-Processor	£89.00

DOCUMENTORS
ASM Flow	£60.00
Clear+ for dBASE	£125.00
Clear+ for C	£125.00
dAction	£55.00
dAnalyst	£139.00
Design/2.0	£215.00
dFLOW	£89.00
Documenter	£179.00
Flow Charting II Plus	£139.00
FoxToolBox	£179.00

EDITORS
BRIEF v.3.0 *NEW*	£159.00
BRIEF v.3.0 *UPGRADE*	£60.00
BRIEF OS/2	£159.00
dBRIEF v3.0 *NEW*	£79.00
BRIEF with dBRIEF	£229.00
C-EDIT	£239.00
EMACS	£199.00
Epsilon	£110.00
ME Editor with C Source	£115.00
Micro ISPF	£110.00
Norton Editor	£49.00
PC/EDIT Plus	£179.00
Personal Rexx	£79.00
PI Editor	£119.00
SPF/PC	£149.00
Turbo Basic Editor Toolbox	£59.00
Vedit+	£115.00

HELP GUIDES AND TRAINING
dBASE on-line	£60.00
Norton Guides for ASM	£75.00
Norton Guides for BASIC	£75.00
Norton Guides for C	£60.00
Norton Guides for OS/2	£89.00
Norton Guides for Pascal	£75.00
Programmer's Library	£239.00

LINKERS & LIBRARIANS
ALINK *NEW*	£195.00
Archive Link	£239.00
Link & Locate ++	£245.00
LinkLok	£245.00
PLINK86 Plus *Now Polytron*	£495.00
PolyLibrarian I	£85.00
PolyLibrarian II	£125.00
.RTLink	£180.00

MAKE UTILITIES
PolyMake v.3.2	£125.00
MKS Make	£99.00
MKS Make 5 User	£320.00
MKS Make 10 User	£515.00

PROFILERS & OPTIMISERS
Codesifter	£75.00
DOSTRACE	£59.00
PC Metric	£120.00

PROTOTYPERS & GENERATORS
Automated Programmer	£605.00
Cogen	£275.00
COMPEDITOR	£110.00
Dan Bricklin Demo II	£125.00
Data Boss for C	£245.00
Data Boss for Pascal	£245.00
Genifer	£239.00
Instant Replay III	£89.00
Matrix Layout v.2.0 (U.K.)	£165.00
Sycero db (U.K.)	£595.00
Sycero db Net (U.K.)	£745.00

SYNTAX ANALYSERS & LINTS
PC Lint v.3.0	£89.00
Pre-C	£150.00

TEXT & UNIX TOOLS
C Programmers Combination	£79.00
C ToolSet	£59.00
MKS LEX & YACC	£165.00
MKS Toolkit v.3.1 *NEW*	£165.00
MKS Toolkit v.3.1 *UPGRADE*	£50.00
PC Fortran Tools	£99.00
PC Yacc	£295.00
PolyAWK	£110.00
PolyAWK OS/2	£115.00

QPARSER Plus	£180.00
Text Management Utilities	£75.00
Turbo Professional	£75.00

TRANSLATORS
BAS_C Commercial	£229.00
Bastoc	£480.00
dBx dBASE to C v.4.7	£389.00
dBx Library Source	£CALL
dBx UNIX/XENIX Source	£CALL
FOR_C v.2.0 MS-DOS *NEW*	£565.00
FOR_C v.2.0 XENIX *NEW*	£630.00
FOR_C++ XENIX/386 *NEW*	£1195.00

VERSION CONTROL
M.K.S. R.C.S.	£125.00
M.K.S. R.C.S. 5 User	£485.00
M.K.S. R.C.S. 10 User	£905.00
PVCS Professional	£395.00
PVCS Network (5 User)	£1395.00
PVCS/PolyMake OS/2	£CALL

BACKUP - RECOVERY - REPAIR
dSalvage	£59.00
Fastback	£105.00
Fastback Plus	£115.00
Mace Utilities	£59.00
Norton Utilities Advanced	£89.00
PC Tools Deluxe	£69.00
XTree Net	£235.00

COMMS. & TERM. EMULATORS
Blast Communications	£149.00
Carbon Copy Express	£199.00
Carbon Copy Plus	£115.00
Crosstalk Mark 4	£145.00
Crosstalk XVI	£115.00
Mirror III	£75.00
PC Anywhere III	£85.00
Sidetalk II	£70.00
SmarTerm	£75.00
SmarTerm 125	£175.00
SmarTerm 220	£115.00
SmarTerm 2392	£135.00
SmarTerm 240	£199.00
SmarTerm 400	£99.00
SmarTerm 4014	£135.00
Z Stem PC-VT100	£59.00
Z Stem PC-VT220	£89.00
Z Stem PC-VT240	£175.00
Z Stem PC-VT4014	£59.00

DISK MANAGERS & OPTIMISERS
HTest/HFormat	£85.00
Mace 5	£65.00
Mace Gold	£95.00
Norton Utilities Advanced	£95.00
SpinRite II	£59.00

OP. SYSTEM ENHANCEMENTS
386MAX	£50.00
386MAX Professional	£85.00
DESQview	£85.00
DESQview/386	£125.00
Norton Commander	£55.00
Omni View	£55.00
PolyShell	£85.00
QDOS II	£45.00
Soft Bytes 386 Max	£45.00
Weiner Shell	£175.00
Power Bundle	£99.00
Windows/286	£59.00
Windows/386	£115.00
X-Tree Pro	£75.00

OTHER OPERATING SYSTEMS
Concurrent DOS-10 User	£175.00
PC-MOS/386	£119.00
QEMM-386	£45.00

FAX OR TELEPHONE CREDIT CARD ORDERS VISA OR ACCESS

PRICING
All prices exclude V.A.T., but include first class recorded delivery.

EDUCATIONAL DISCOUNTS
Educational discounts available on certain products.

CATALOGUE & FACT SHEETS
Call for our new Summer '90 catalogue, over 135 pages of Software Development Tools.

PCW 01/07/90

THE SOFTWARE CONSTRUCTION CO. LTD.

1 THE MALTINGS, GREEN DRIFT, ROYSTON, HERTS. SG8 5DB (0763) 244114 FAX. (0763) 244025

CIRCLE NO. 159

Could YOU take the STRAIN?

You are reading .EXE Magazine, so you don't have to be told about this Magazine's standards. Our writers really have to know their stuff. We require a technical depth of knowledge, a degree of insight and a conformance to an especially readable and articulate house style that puts (as Alan Whicker might say) 'certain other magazines that we could mention' to shame. But all this is not enough.

If you write for .EXE, you will also have to cope with the corrupting influence of vast wealth in the form of your writer's fee (often running to as much as £100!) and the frightful burden of international fame. When you publish in .EXE, we regret that we cannot guarantee to prevent the media circus from hounding you, interviews on breakfast television several days a week, perhaps even hoards of screaming fans pursuing you everywhere you go.

If you think that you are tough enough to take this sort of thing, take a gander at these upcoming special issues:

- *October - Security and software protection*
- *November - Object-oriented programming*
- *December - Debugging techniques*
- *February - C and C++*
- *Plus a Real Time supplement with November's issue.*

If you have expertise in one of these fields, and feel that you could handle Life at the Top, we would like to hear from you. Why not jot down a summary of your proposed article, and send it to the address below. Or, if you are not sure how to begin, write for a set of Contributors' Notes, to help you get going. After all, even Verity had to start somewhere.

The Editor, .EXE Magazine,
10 Barley Mow Passage,
London W4 4PH.

PS - If you have an idea for a brilliant .EXE article, but it does not fall into one of the above categories, write and tell us about it anyway.

.EXE

G.W. Computers Inc.
ADCI, 4 Eagle Square,
E.Boston. MA02128
U.S.A.
Tel: **617-569-5990**
Fax: **617-567-2981**

System-IX (Networks) Ltd.
55, Bedford Court Mansions,
Bedford Avenue, London.
WC1B 3AD U.K.
Tel: **071-636-8210**
Fax: **071-255-1038**

Menu-IX System-IX DBMS-IX Sysix AI Spell Merge Look GURU Crypt Tawk Tgrep

Hardware System
386-16 SX starting from £895.00 to
486-25 w: 4mb ram,VGA,.80mb disk £3750.00
call for quotations and latest prices.

SCO UNIX / XENIX

	286	386
SCO Xenix Op Sys	£357	£422
SCO Xenix Dev Sys	£357	£455
SCO Xenix Text Pro		£126
SCO Xenix Comp Sys	£775	£905
SCO Unix Op Sys		£525
SCO Unix Devel Sys		£613
SCO CGI Graph Run/Sys	>	£191
SCO Multiview	£256	£321
SCO VP/ix		£321
SCO Foxbase+	£516	£647
SCO Foxbase+ Run/Sys	>	£191
SCO Masterplan	>	£516
M/Soft Word v.5.0	>	£563
Multiplan	>	£321
SCO Eroff Documentor	£451	£516
SCO Lyrix v6.0	>	£386
SCO Professional	>	£386
SCO Image Builder	>	£386
M/soft BASIC compiler	>	£451
M/soft BASIC interp.	>	£256
M/soft FORTRAN comp.	>	£451
M/soft PASCAL comp.	>	£451
SCO XENIX-NET Server	£386	£451
SCO XENIX-NET 3 Server	£841	£1036
SCO TCP/IP (Req streams) Run/Sys		£227

OPEN DESKTOP and XSIGHT also available. £call
Complete systems (runtime/dev) include MENUIX FREE.

INTERACTIVE 386/ix UNIX

New Rel 2.2		Application	Network	Workstation
Platform	(2 User)	£495	£695	£855
	(Multi)	£1095	£1350	£1495
Developer	(2 User)	£1095	£1155	£1295
	(Multi)	£1750	£1815	£1945

These are VERY powerful professional systems, For an Application starter you need about 12-20 mb of hard disk, but a workstation developer needs hard disk space to install 65 by 1.2 or 1.44 mb diskettes. To include things like: Op'sys, Soft'dev (LEX,YACC,SCCS etc), Standard C, New C, Text prc (NROFF,TROFF,TBL,etc), XII, XII Dev, Vpix (DOS SHELL), TCP/IP,NFS, Online MANS, Looking glass, Viewten UI, Codewatch, Co-Edit. Full documentation weighs in at about 40 lbs. All modules can be purchased separately.

Some Parts	1-2 Users	Multi User
386/ix Runtime	£255	£495
Software Development System		£495
X11.3 Runtime	£240	£265
X11.3 Development System	£580	
Multiview		£395
Motif Development System	£365	
TCP/IP		£355
NFS	£355	£420
TCP/IP + NFS	£425	£585
PC Interface, DOS Bridge Module		£365

System kits over £900 include MENUIX. FREE.
System kits over £1500 include SYSIX and GURU FREE.

Microport UNIX

		286	386
Complete System		£395	£495
Comp. Sys. Unlimited		£538	£718
Runtime System (2 user)		£159	£189
Run. Sys. Unlimited		£299	£358
Unlimited User Upgrade		£159	£189
Software Development	£189	£329	
Text Preperation		£129	£129
DOS Merge Limited		£165	£249
DOS Merge Unlimited		£165	

Additional HOT Items

Q-Calc spreadsheet	£695
Master-Plan project management	£1295
Uniplex business softwaare	£ 1150
PC-CONNECT	£450
SQL-Connect	£450
PC-TCP	£375
PC-Xview	£245
Xvision	£295
Emacs	£325
Designer C++	£1195
Terminal Control for UNIX AND DOS	£695
Menuix	£49.95
Sysix	£99.95
Sysix + Guru	£145
System-IX	£295
Sysed-IX	£295
Cshell-IX for DOS	£50

We Stock **Hundreds Of DOS** Products From **ALDUS** To **ZSOFT** With **LOTUS, MICROSOFT & SAGESOFT** In Between !

Exceptional products : Entire Shakespeare works, Theasaurus & Dictionary, Bibles, 500,000 word lists, Moby Words,CD-ROMS, C sources, etc.

We also sell CD-ROM drives and WORM drives, Network cards, specialist printers and plotters, Laser printers and colour thermal printers (QMS).

*** Brand NEW ***
Portable hard disk technology. FITS IN A STANDARD 5.25" DRIVE BAY.
FRAME & REMOVEABLE DRAWER, enables you to fit a 3.5" drive of 40-200 mbytes into the draw, and remove it in 1 second for safeguard or even better, to just plug in any O/S of your choice, or take your file-systems from machine to machine. JUST £55

Cut More Term Cd Where Sort Grep Idle CB Trees Evaluate Awk Banner C Source Fgrep Help Head Tail

Xenix Xenix Net Unix Interactive System V TCP/IP DOS Token-Ring OS/2 Ethernet Novell Lan Mic Net Aix VP-IX VMS QMS

Compaq Toshiba IBM Epson Qume NEC Seagate Wyse Relysis Zenith Priam Ranger

CIRCLE NO. 160

object code was the real challenge and the syntax of the language was very much a secondary issue.

Figure 1 shows an example program taken from the original FORTRAN manual produced by IBM and dated October 1956. Program statements were punched one to a card, with fixed fields for the statement number, the FORTRAN statement itself, and the card sequence number (in columns 73-80). This format has been carried down right to the present day, although virtually no-one punches programs on cards any longer.

Figure 2 shows the .EXE Triangle problem coded in the original FORTRAN for the IBM 704 computer. Perhaps the most remarkable feature of this program is that it needs just one correction to make it compile and run perfectly on a present-day FORTRAN system. (The correction is the deletion of the `FREQUENCY` statement - of which more later. My compiler also issued a 'warning' diagnostic because there was no `END` statement - a language refinement that did not, in fact, appear until FORTRAN II in 1958.) On the other hand, the program itself does not look very much like a 'modern' FORTRAN program; this is because it is written using the rather primitive statements available in the original 1957 FORTRAN; these statements still exist in the language for backwards compatibility, but some of them are now known as 'deprecated features' that are candidates for eventual removal from the language.

FORTRAN recognised two types of variable: fixed point (or index) and floating point. (In the 1966 FORTRAN standard, these became known as `INTEGER` and `REAL`.) Variables names could be a maximum of six characters in length. Why six? For the simple reason that the IBM 704 stored six 6-bit characters in its 36-bit word, and storing one identifier per word made table look-up easy and efficient. This is a restriction that has carried right down to the 1977 standard. Variables were not declared, but were implicitly typed according to the initial letter (`I` to `N` for fixed-point variables, and the remaining letters for floating-point variables). The implicit naming convention remains one of the most controversial features of FORTRAN, since simply misspelling an identifier can result in the creation of a 'phantom' variable. It was this particular feature that was (unfairly) blamed for the Mariner disaster, and became something of a *cause célèbre* for the use of programming languages with strongly-typed variables.

A major attraction of FORTRAN was its powerful arithmetic expressions, which accepted the five arithmetic operators and permitted parentheses nested to any depth. The important point, however, was that the translator could not only compile the expressions, but it would also produce virtually optimum code - for example, evaluating common sub-expressions only once.

Arrays were declared using the `DIMENSION` statement and could be one, two or three dimensional - a particularly advanced feature at the time. Closely allied to the use of arrays was the `DO` loop control statement. Looking very like the `FOR` statement of later languages, the control variable and its increment had to be of fixed-point type; this limitation was because, for efficiency reasons, the control variable was mapped onto an IBM 704 index register which stored only integer values.

The `IF` statement also reflected the underlying IBM 704 instruction set, which contained condition-branch instructions that tested whether a value was zero, positive or negative. Thus the statement IF (A+B) 10, 21, 15 would evaluate the expression A+B and: branch to 10 if the result was negative; branch to 21 if the result was zero; or go to 15 if it was positive. In the present age of goto-less programming, the arithmetic `IF` statement has largely fallen into disuse; and although it remains in the current FORTRAN standard for compatibility, it is considered a relic and its use is regarded as poor programming practice.

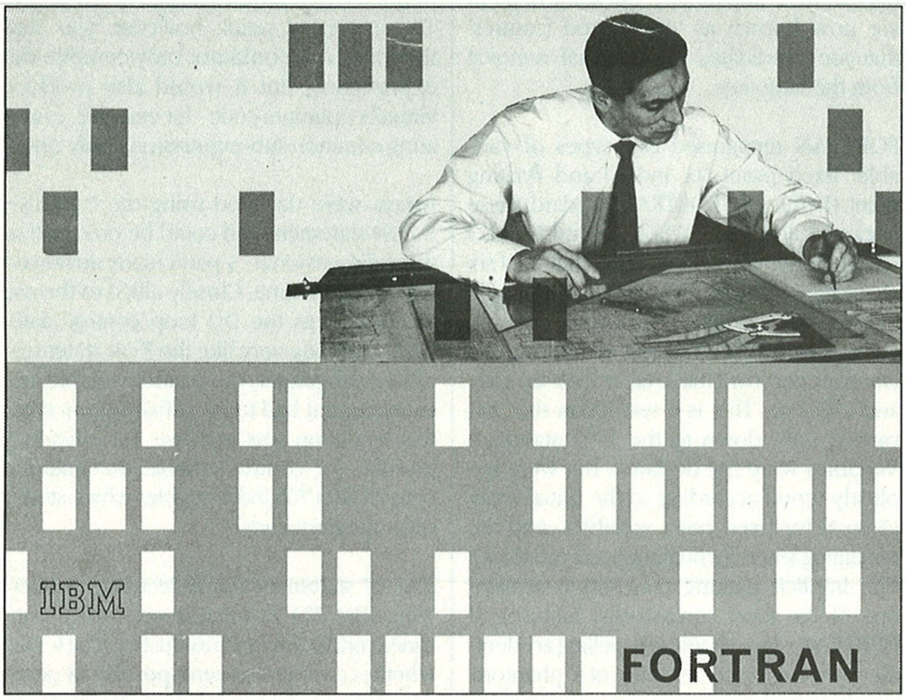
The input-output statements have much in common with the current standard, and are essentially a subset of it. One curiosity is the way of defining a string in a `FORMAT` statement, where a string such as `'TRIANGLE'` is written `8HTRIANGLE` (where 8 is the string length and H stands for the Hollerith punched-card code). The reason for this oddity is that the original IBM 026 card punch, used for preparing FORTRAN programs, did not have a quote mark in its character set. Again, this notation for defining string constants remains part of the FORTRAN standard for backwards compatibility.

The STOP statement is an interesting harkback to the days when FORTRAN programs were run in uni-programmed environments. The octal constant following the stop-statement (eg STOP 70707) would be displayed on the IBM 704 console display lights - this was useful for identifying error halts. A variant of the stop statement, PAUSE behaved similarly - but the program could be restarted by pressing the appropriate button on the console. This was useful for synchronising a program with its input-output (for example, halting the program to lead decks of data cards, or to change magnetic tapes),

```
C
C       THE TRIANGLE PROBLEM
C
        FREQUENCY 5(3)
        DIMENSION T(3)
        READ 100, T(1), T(2), T(3)
  100   FORMAT (3F6.2)
    5   DO 1 I = 1,3
    1   PRINT 101, T(I)
  101   FORMAT (3F6.2)
C
C       INEQUALITY TEST
        IF (T(1)-(T(2)+T(3))) 11,11,13
   11   IF (T(2)-(T(3)+T(1))) 12,12,13
   12   IF (T(3)-(T(1)+T(3))) 15,15,13
   13   PRINT 102
  102   FORMAT (22HTHIS IS NOT A TRIANGLE)
        GOTO 2
C
C       EQUILATERAL TEST
   15   IF (T(1) -T(2)) 20,16,20
   16   IF (T(1) -T(3)) 20,17,20
   17   PRINT 103
  103   FORMAT (31HTHIS IS AN EQUILATERAL TRIANGLE)
        GOTO 2
C
C       ISOSCELES TEST
   20   IF (T(1) -T(2)) 21,23,21
   21   IF (T(2) -T(3)) 22,23,22
   22   IF (T(3) -T(1)) 25,23,25
   23   PRINT 104
  104   FORMAT (29HTHIS IS AN ISOSCELES TRIANGLE)
        GOTO 2
C
C       TRIANGLE MUST BE SCALENE
   25   PRINT 105
  105   FORMAT (26HTHIS IS A SCALENE TRIANGLE)
    2   STOP 70707
```

Figure 2 - The Triangle Problem

Figure 3 - An IBM brochure for FORTRAN, circa 1960



and for tracing program errors. Again, the STOP and PAUSE statements are still part of the standard, and can be used meaningfully in conversational FORTRAN systems.

Finally, the FREQUENCY statement. The original purpose of this optional statement was to enable the programmer to give the compiler a hint that helped it to optimise the object program. For example, the statement FREQUENCY 10 (100), 21 (1, 2, 1) would mean: DO statement number 10 will be obeyed 100 times on average; and the three destinations of IF statement 21 will be taken in the approximate ratio 1:2:1. The compiler could make use of this information to fine-tune the object code. This was an important feature when FORTRAN was trying to gain credibility and efficiency was paramount; but later the FREQUENCY statement was ignored by most compilers and it did not appear in the 1966 standard.

The rise and rise of FORTRAN

FORTRAN was released in April 1957, to a largely sceptical user community. But when those same users discovered that FORTRAN could and did produce object programs 90% as good as hand-coded assembly language programs, the scepticism disappeared. Soon, many installations were spending half of the available machine time running FORTRAN programs and, by Autumn 1958, some 60 installations were using the system. The language proved to be an amazing productivity lever: a programmer could be trained in about 20 hours (compared to the weeks it took to get proficient at machine code); and, while programmers still tended to produce about the same number of debugged program statements per day, the fact that each FORTRAN statement was worth four or five machine-code statements meant that total output was increased by the same factor.

In IBM, scepticism towards FORTRAN now changed to positive support. It became a serious selling point and brochures and publicity material were produced (Figure 3). And as FORTRAN gained official product status, the project was transferred from the Programming Research Group to the Applied Programming Department, which took over field support and maintenance. Work also started in 1958 on a successor, FORTRAN II.

One of the major defects of the original FORTRAN had been that a user's program had to be completely recompiled every time it was corrected - and as a typical compilation took several minutes, this made program debugging very expensive. To improve on this, FORTRAN II introduced program units and subroutines (with corresponding END statements) and the concept of a linkage-editor. Now user programs and subroutines could be compiled just once and were subsequently linked together to produce a binary program. A consequence of this efficiency improvement was the development of large-scale, portable libraries of pre-compiled subroutines for matrix work and other numerical procedures, which further consolidated FORTRAN's popularity. The source-object-binary model of program organisation is today one of the universal characteristics of programming systems. It is not widely appreciated that its origins are in FORTRAN II.

In spite of the early success of FORTRAN, in the late 1950s its long-term future was far from assured. There was a major international effort to design and promote Algol 60 as a new world-wide standard for scientific programming; in fact, John Backus took a leading role in the Algol group and he had no commitment whatsoever to FORTRAN. However, during the period 1957-60 when Algol was being designed, US computer users took to FORTRAN so whole-heartedly that they began to demand that other manufacturers produce compilers compatible with IBM FORTRAN. Thus FORTRAN had already become a *de facto* standard by the time that the Algol 60 Report was published, so that in America the latter was never really a serious contender.

In the early 1960s, FORTRAN compilers from different manufacturers proliferated, but the lack of any sort of a FORTRAN standard meant that different implementations differed in small but important ways. It was to address this problem that the American Standards Association (ASA - later ANSI) established a FORTRAN committee in May 1962. The first FORTRAN standards appeared in 1966, which defined two versions of the language: Basic FORTRAN, roughly corresponding to IBM's FORTRAN II; and FORTRAN, corresponding to what was then IBM's latest offering, FORTRAN IV.

Standards committees grind exceedingly slow, and more than a decade was to pass before the next standard - FORTRAN 77 - was to emerge. The 1977 standard incorporated new control statements for structured programming which had come to the fore in the late 1960s and early 1970s; and there were features such as recursion, and support for non-numerical computation, time-shared operation, and direct-access devices. It says something for the adaptability of FORTRAN that it could take on-board all these new ideas and yet maintain almost total compatibility with the earlier FORTRANs.

The latest standard, FORTRAN 8X, of which few implementations yet exist, will take the language through the 1990s and into the next century. Years ago, Tony Hoare stated, 'I don't know what the language of the year 2000 will look like, but I know it will be called FORTRAN.' Well, FORTRAN 8X shows us what it looks like: it is a FORTRAN for today, but also one that is compatible with, and looks very much like, the FORTRANs that went before. The more FORTRAN changes; the more it stays the same.

.EXE

Dr Martin Campbell-Kelly lectures in computer history at Warwick University.

MULTIPLE INTELLIGENT COM PORTS FOR OS/2

THE IMPAC 8 INTELLIGENT COMMS CARD

* MICROSOFT COM X COMPATIBLE
* 8 TO 32 FULL MODEM PORTS
* REDUCE CPU OVERHEAD FROM 40% TO 1% PER PORT
* FITS ALL ISA (PC/XT/AT) MACHINES
* DOES NOT USE SHARED RAM
* PROVEN FIELD PERFORMANCE
* U.K. DESIGN AND MANUFACTURE
* DRIVERS FOR OS/2, UNIX, XENIX

DEALER ENQUIRIES WELCOME

For more information contact:
SCL, Downswood, Carlton Road
Reigate, Surrey RH2 0JQ
Telephone: 0737-762200

DOS DRIVER ANNOUNCED

CIRCLE NO. 161

FORTRAN

FTN77 - Power Compilers

The Salford FTN77/386 and 486 compilers use DOS extender technology to turn your 386sx, 386 or 486 based PC into a powerhouse. FTN77 programs run under MS-DOS, alongside your existing applications, but what a difference! Execution speed on a 486 is comparable to a top-of-the-range VAX 8650, and compilation is even faster!

- No 640K limit. Accesses a linear ***4 Gigabyte*** address space.
- Virtual Memory support.
- Supports Intel and/or Weitek numeric coprocessors.
- Emulates 80387 if no coprocessor available.
- Lightning fast Execution. Uses the full 32 bit 80386 instruction set.
- Lightning fast Compile and Link. Nothing else comes close...
- ANSI validated by the NCC and US Dept of Commerce.
- Many extensions (e.g. F66 compatibility, inline assembler)
- Extensive run-time library including many system facilities.

FTN77 is more than just a number-cruncher - it's a ***bug-cruncher*** too! The integrated source level debugger combines remarkable power with an intuitive window based user interface. Just as importantly, the optional run-time checks on interfaces and the use of uninitialised data detect bugs that very few other compilers or analysis tools can spot.

FTN77 is compatible with numerous graphics packages, including the powerful and portable INTERACTER library from ISS. Please call for details and a ***free*** demo disk.

"For sheer speed and sleekness, the (Salford) compiler stands out like a Ferrari among Yugos".
(Computer Language Magazine, May 1990).

Polyhedron Software Ltd., **Magdalen House,**
98 Abingdon Road, Standlake, Witney, OX8 7RN, England
Tel (+44) 0865-300579 Fax (+44) 0865-300232

Polyhedron Software

All trademarks acknowledged

CIRCLE NO. 162

FTN77/486

Ready when you are!

- ☐ **The FASTEST PC Fortran compiler around.**
- ☐ Compiled programs exploit the 486 hardware using paged virtual memory giving an address space of up to 4 Gigabytes.
- ☐ ANSI Fortran 77 plus mainframe extensions.
- ☐ Runs under MS-DOS. FTN77/486 is supplied complete with our DBOS/486 DOS extender.
- ☐ Optional global optimisation or checkout mode with diagnostics.
- ☐ Full window-based source level debugger.
- ☐ Provides a library of over a hundred routines for MS-DOS access, non-standard file manipulation, window management, graphics, etc.
- ☐ Supported by experts and supplied with a comprehensive, fully indexed reference manual.
- ☐ Allows libraries to be linked dynamically to programs during execution. Dynamic linking means that EXE files are kept small.
- ☐ Contains a full 32-bit in-line symbolic assembler with window-based debugger.
- ☐ Interface to real mode libraries.
- ☐ Selection of graphics libraries available.
- ☐ Optional Weitek co-processor support.

FTN77/386 AVAILABLE NOW!

SALFORD SOFTWARE MARKETING LTD
COMPUTING CENTRE · UNIVERSITY OF SALFORD · SALFORD · M5 4WT · ENGLAND
Telephone (+44) 061 745 5678 Telefax (+44) 061 745 5666

*MS. DOS is a trademark of Microsoft Corporation. IBM and PS/2 are trademarks of the International Business Machines Corporation. DBOS/486, FTN77/386, FTN77/486 are trademarks of the University of Salford.

CIRCLE NO. 163

G.W. Computers Inc.
ADCI, 4 Eagle Square,
E.Boston, MA02128
U.S.A.
Tel: 617-569-5990
Fax: 617-567-2981

System-IX (Networks) Ltd.
55, Bedford Court Mansions,
Bedford Avenue, London.
WC1B 3AD U.K.
Tel: 071-636-8210
Fax: 071-255-1038

Menu-IX | System-IX | DBMS-IX | Sysix | AI | Spell | Merge | Lock | GURU | Crypt | Tawk | Tgrep

Cut | More | Term Ctl | Where | Sort | Grep | Idle | CB Trees | Evaluate | Awk | Banner | C Source | Fgrep | Help | Head | Tail

Xenix | Xenix Net | Unix | Interactive | System V | TCP/IP | DOS | Token-Ring | OS/2 | Ethernet | Novell Lan | Mic Net | Aix | VP-IX | VMS | QMS

MenuIX, SysIX & SystemIX under Xenix,Dos & Unix.

Main Options

Administration | Files | Backup | Maintenance | System | Acct | Mail

Ad Options
C Shell
Bourne Shell
Visual Shell
Shutdown
Startup
Install Applic
Remove Applic
Display Process
Display TTY
Modify TTY

File Options
Free Space
Maintenance
Create Dir
Remove Dir
List Dir
Print Dir
Create File
Remove File
Print File

Back Options
Backup Files
Backup Filesystem
List Backup
Print Backup
Restore Files
Format Diskette
Copy Diskette

Mnt Options
User
Printer

Invoice
Letters

Acct Options
Account User
Account Group
Account TTY
Account All
Enable Account
Disable Account

Mail Options
Phone To User
Send Mail
Read Mail
Write All
UUCP

Menuix from
£49.00

User Options
Create User
Remove User
Create Group
Remove Group
Change Password
Applications

Print Options
Add Printer
Remove Printer
Enable Printer
Disable Printer
Start Schedule
Stop Schedule
Lp Status
Print File
Set Default Lp

App Options
Spreadsheet
Wordprocessor
Database
Other

Db Options
Sys-IX
System-IX
Guru
Csh-IX

UUCP Options
UUCP Setup
Dialcode
System
Devices
UUCP Stat
Clear Log
Remove Log

Scrollable-resizable Help

Everthing onscreen is runtime interpreted from your own edited ascii file
Online Hot-Key dynamically scrolls moveable context help from your ascii file
Unlimited chaining through pull-down menues with ITEM status line help
Supports full system and multiple system calls, auto-minimise to 7k under DOS
256 colours, highlight bars, unavailable items. All formats XENIX,UNIX,DOS
Edit ascii file to translate entire runtime to any European language
Runs child system calls and script files. 120 page manual. Unbeatable value

Sysix ......From
£99.00 to £295.00

Powerful Data Entry

Sys - IX data entry is easy to use for the new user and powerful for the experienced professional. A multi-level design gives high level functions ease of use leaving lower level functions the versatility you will demand. This makes Sys IX perfect for users of all levels of expertise for any application.

Input formatting with pictures, Complete colour control for inputs, Powerful validations , Pick from selection sets, Definable input control keys , Security level for each field or screen Different status line for each field, Right ot left numeric input, Floating dollar signs Commas inserted into numeric fields, Insert or overstrike mode, Pre-function or post-function for any field, Field specific help messages, Automatic scrolling fields Abort screen without changes, Option for automatic carriage return, Optional field blanking on first keystroke program accessible for customising,Single field or full screen input, Move field next, previous, up, or down No. fields/scr limited only by RAM, Input masking, Change one field based on another Skip fields .

Fast Windows

Window management is totally automatic with Sys - IX. After a window is opened Sys - IX will allow you to print to any window, at any time, without problems. No structures to fill, no questions to answer - just open a window and all output will be directed to that window - it's just that simple.

Automatic collision protection, Printing to any window, Bordered or borderless windows Runtime adjustable help windows , Number limited only by RAM, Move or reshape windows under program control Interfaces to all data entry windows, Zoom in or zoom out windows, Pull-down windows Vertical or horizontal scrolling, Choice of borders, Pop up Windows irtual screens automatically scroll, Advanced screen optimization, Unix extensive Termcap file for extra features.Supports all terminals ,Supports colour , Supports line graphics for windows, lines, and boxes plus optional scroll bars Extremely portable , Auto. restricts I/O in window Auto-select colours or manual , Print to background and foreground window at the same time

Includes POWERFUL Data Transfer & Reporting Utilities on full system

System-IX INCLUDES : GURU, CRYPT, AI, BTGURU, PLUS SEVERAL APPLICATION PACKAGES.
UNDER DOS, also includes: CSHELL, AWK, GREP, FGREP, DIFF, CAL, HEAD,TAIL, MORE,CAT, TOUCH, TEE, SORT, FIND, DETAB, ENTAB, ENCRYPT,DECRYPT: (Subject to variation without committment).

Compaq | Toshiba | IBM | Epson | Qume | NEC | Seagate | Wyse | Relysis | Zenith | Priam | Ranger

CIRCLE NO. 164

No games until lunchtime

TIMELOCK is a utility by Robert Schifreen that allows you to restrict the days and times at which specific MS-DOS programs may be run.

Large computer systems come with clever utilities to prevent certain programs being run outside certain times of the day, or days of the week. It's possible to write such a program for the PC, and the TIMELOCK.ASM listing that you see here is my attempt to do just that. It's not foolproof, and anyone with a reasonable knowledge of MS-DOS will probably be able to get around it. However, if you look after a large department of computer-illiterate users, and you want to prevent them playing games except during the lunch hour or at weekends, then this utility is better than nothing. If you wish, you can also use it to place certain programs completely 'out of bounds'.

How It Works

Like most of the listings that find their way into the Code Page, TIMELOCK is a TSR. This time, we sit on Interrupt 21h, function 4Bh. This is the EXEC call which is used by MS-DOS to load and execute a program. Suppose a user runs a game program by typing INVADERS at the DOS prompt. COMMAND.COM then searches the current directory, and/or the path, for a file called INVADERS with an extension of .COM, .BAT or .EXE. Assuming that it finds the file, it calls DOS function 4Bh, with the full path name and filename of the file. For example, if you type INVADERS at the DOS prompt, the string that eventually gets passed to function 4Bh may be 'C:\ROBERT\GAMES\INVADERS.EXE'.

If all goes well, function 4Bh executes the file, and then returns with the carry flag clear. This

```
; TIMELOCK.ASM - v1.0, June 1990. Robert Schifreen.
; ------------------------------------------------

    code   segment
    assume  cs:code,ds:code,es:nothing
    org    0100h

start:

    jmp   init    ..      ; Installation code is at the end

intent:              ..   ; Come here on every Int 21h call

    pushf
    cmp   ah,04Bh         ; Is this an EXEC call?
    je   check_time       ; Jump to time-checker if it is
    jmp   intxeq          ; Else, do nothing

check_time:

                          ; An EXEC call is in progress. Get
                          ; name of file that's the subject
                          ; of EXEC call. Path name of file
                          ; is ASCIIZ, in DS:DX.

    push  ds
    push  bp
    push  di
    push  si
    push  ax
    push  bx
    push  cx
    push  dx

    mov   bp,dx

go_again:

    mov   ah,byte ptr ds:[bp]  ; Get char from ASCIIZ filename
    cmp   ah,0                 ; Are we at the end of the string?
    je   at_end                ; Jump if yes
    inc   bp                   ; Else increment pointer
    jmp   short go_again       ; And go get the next character

at_end:

                  ; BP now points to trailing zero. Work
                  ; backwards to find a backslash.

    mov   ah,byte ptr ds:[bp]  ; Get char
    cmp   ah,05Ch            ; Is it a backslash?
    je   backslash           ; Jump if found
    dec   bp                 ; Else, decrement counter...
    jmp   short at_end       ; ...And go round again

backslash:

                      ; DS:[BP] is the start of the ASCIIZ
                      ; filename, rather than a full pathname
                      ; with directory and drive info.
                      ; Search list of time limit records to
                      ; see if there's an entry for the file.

    mov   di,offset cs:limits  ; Start of time limit data
    inc   bp              ; Skip backslash and point to 1st char
    mov   si,bp           ; Keep a copy of BP

find_match:

    mov   ah,byte ptr ds:[bp]  ; Get char from filename to be EXECed
    mov   al,byte ptr cs:[di]  ; Get char from file in limits data
    cmp   ax,0            ; If both are zero, match was found
    je   match_found
    cmp   ah,al           ; Are both chars the same?
    jne   no_match        ; No, then strings don't match,
    inc   bp              ; Chars the same, so the strings...
    inc   di              ; ...Match so far. Keep checking.
    jmp   short find_match

no_match:

                      ; Current string does not match.

    mov bp,si         ; Reset BP to start of EXECing filename

                      ; Is there another file name in the
                      ; Time limit data records?
                      ; If so, see if it matches.

    inc   di

find_end:

    mov   al,byte ptr cs:[di]  ; Find the end of the current filename.
    cmp   al,0
    je   got_end
    inc   di
    jmp   short find_end

got_end:

                        ; DI points to the trailing zero of a
                        ; Limits filename. Skip the time data
                        ; And see if there's another filename.

    add   di,49         ; Skip 48 bytes of time data
    cmp   byte ptr cs:[di],0   ; Are we on an end_of_records marker
    je   no_more
    jmp   find_mat      ; Go look for another match

no_more:

                        ; The string does not exist in the
                        ; List of prohibited files, so
                        ; Execute it as normal.
```

Figure 1 - TIMELOCK.ASM

tells the caller that the program was successfully executed. If for some reason the program cannot be executed, function 4Bh returns with the carry flag set, and an error code in AX. One such error code is Access Denied, which is error code 5. This is returned if, for example, a file cannot be loaded because it is already in use somewhere else on a network. COMMAND.COM then displays the error message.

TIMELOCK is very simple. It maintains a list of files that may not be run and, if you attempt to run one of these, at a forbidden time of day, it returns an error code 5 to the routine that called function 4Bh, instead of loading and executing the file. This means that the user will see an error message Access Denied if he tries to run a program at a forbidden time.

Of course, there's an easy way around this. The user could simply rename the file, to a name that doesn't appear in the forbidden list. The user could also boot from a floppy disk, to avoid the TIMELOCK program being run at all, or change the system time and date. The first situation is easy to get around - use something like DDC to prohibit use of the RENAME command. (DDC is a utility to disable the use of internal DOS commands, and appeared in a recent Code Page article). You could, if you wish, write a special TSR that refuses to let the user rename files whose names are the subject of a TIMELOCK forbidden list.

As it stands, TIMELOCK is deliberately unfriendly. It doesn't print a message when it loads, so the user will not easily be able to tell that it is in force. Also, I haven't included any easy way of amending the list of files, or the times at which the files can be run. To configure these, you have to edit the source code and reassemble. This again is a deliberate piece of hostility.

The format of the data area is quite straightforward. Each data record starts with a zero-terminated filename. This is the name of the file whose use you wish to restrict. Following the name of the file are 48 data bytes. The first 24 of these govern the use of the file during weekends, ie when the system day-of-the-week call returns Saturday or Sunday. The remaining 24 bytes govern the use of the file on weekdays. Each of the 24 bytes relates to one specific hour as follows: if byte n is set to 1, then use of the file is permitted when the current hour is n. If the byte is zero, then use of the file is not permitted.

So, to forbid the use of a file between 1pm and 2pm, you need to set the 14th byte (byte 13) to 0. This locks the file between 13:00 and 13:59, which is near enough. You can have as many records as you like, as long as the size of the .COM file doesn't exceed 64 KB. Just remember to add the terminating zero after the last record.

Incidentally, I could easily have bit-mapped the data area. To hold data for 24 hours takes only 24 bits, which fit snugly into 3 bytes. However, unless you have a large number of forbidden files, the extra code required to manipulate the bitmaps means that you don't save much memory at all. And the coding would have been harder, too.

EXE

```
    pop   dx
    pop   cx
    pop   bx
    pop   ax
    pop   si
    pop   di
    pop   bp
    pop   ds

    jmp   intxeq

match_found:

    inc   di

                        ; The file that DOS wants to EXEC is
                        ; The subject of a timelock.
                        ; CS:[DI] is the first byte of the
                        ; file's 48 timelock data bytes.
                        ; The first 24 bytes are for Sat & Sun.
                        ; The next 24 are Mon-Fri.

    mov   ah,2Ah
    int   21h              ; Get day of week into AL
    cmp   al,0             ; Is today Sunday?
    je    weekend          ; Jump if so.
    cmp   al,6             ; Is today Saturday?
    je    weekend          ; Jump if so
    add   di,24            ; Today is not weekend, so skip the
                           ; weekend data bytes.

weekend:

                        ; CS:[DI] now points to the start of
                        ; 24 bytes that govern today's rules
                        ; for the execution of the file.

    mov   ah,2Ch           ; Get the current hour
    int   21h              ; Get hour (0 - 23) in CH.
    mov   cl,ch            ; Put hours into CL
    mov   ch,0             ; Zero CH, to make CX = hours
    add   di,cx            ; Point to correct data byte for
                           ; Current hour.

    cmp   byte ptr cs:[di],1    ; Is execution permitted?
    je    permitted

                        ; Execution forbidden. Set carry flag
                        ; to tell DOS that file can't be
                        ; executed.

    pop   dx
    pop   cx
    pop   bx
    pop   ax
    pop   si
    pop   di
    pop   bp
    pop   ds

    popf
    mov   ax,5             ; Return "Access Denied" error to DOS.
    stc                    ; Set carry flag to indicate error

    retf  2                ; And we're all done. IRET instead of
                           ; Chaining to Int 21h func 4Bh handler.
                           ; Use RETF 2 instead of IRET, to avoid
                           ; corrupting flags register.
permitted:

    pop   dx
    pop   cx
    pop   bx
    pop   ax
    pop   si
    pop   di
    pop   bp
    pop   ds

intxeq:

    popf                   ; Retrieve flags from stack.
    db    0EAH             ; Absolute JMP (segment + offset)
intip  dw   0              ; Filled by init code
intcs  dw   0              ; Filled by init code

                           ; Start of data for list of files.


limits db 'DEBUG.EXE',0               ; Filename
    db 1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 ; Sat & Sun
    db 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 ; Mon-Fri

    db 'DEBUG.COM',0                  ; Filename
    db 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 ; Sat & Sun
    db 1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 ; Mon-Fri

    db 0                              ; No more data


                                      ; Install TimeLock program.

init:

    push  ax
    push  es
    mov   es,cs:[2ch]      ; get seg adr of our env block
    mov   ah,49h           ; free a block
    int   21h              ; free the block
    pop   es
    pop   ax

    mov   ax,cs            ; Set DS = CS
    mov   ds,ax
    mov   ah,35h           ; Get old interrupt vector
    mov   al,21h
    int   21h
    mov   intcs,es         ; Save in long JMP
    mov   intip,bx
    mov   ah,25h           ; Set new interrupt vector
    mov   al,21h
    mov   dx,offset intent
    int   21h
    mov   ax,3100h         ; Termination function
    mov   dx,(init-start)/16+17
    int   21h              ; Terminate and stay resident.

    code  ends
    end   start            ; End of TIMELOCK.ASM
```

Figure 1 - TIMELOCK.ASM (Continued)

EVERLOCK

COPY PROTECTION

Designed for clone compatibility & strength

- **Thwarts ALL Bit-copy Software**
- **Protect any COM/EXE w/o Source changes**
- **Shut down Debug Tracing & Disassemblers**
- **Supports all Hard & Floppy Disk formats**
- **Restrict or allow Network Installs**
- **Limit program use by Date or # Executes**
- **Remotely reset Installs, Date or # Executes**
- **No damaged media or I/O plugs**

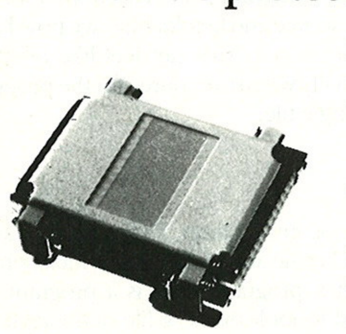
For IBM or Clones £465.00 + p&p £6.50
FREE INFORMATION & DEMONSTRATION
Limited Use Version £150.00 + p&p £6.50
(Prices exclude VAT)

USER FRIENDLY micro-computer systems

ICS House, Edward Street
Redditch, Worcs B97 6HR
Tel: 0527 585550 Fax: 0527 65111

CIRCLE NO. 165

SOFTLoK™ & SOFTLoK PLUS™ software protection systems



- For IBM PC, PS/2 and compatibles
- Installed in seconds in the parallel printer port
- Assembler & high level source code provided
- Fast response
- Transparent to printer
- Small physical size

SOFTLoK

SOFTLoK units contain a code that can be read and verified by routines provided in assembler and a variety of high level languages.

Price: 10-99 units £12.00, 100+ units £8.00
Evaluation kit £20.00

SOFTLoK PLUS

SOFTLoK PLUS units contain a small area of non-volatile memory protected from unauthorised reading or writing by a password. Both the memory area and the password can be changed at any time by the routines provided in assembler & several high level languages. As the memory can even be altered in the field it can contain counters & expiry dates as well as serial numbers etc.

Price: 10-49 units £40.00, 50+ units £35.00
Evaluation kit £40.00

SOFTLoK International Limited
Tel: (0254) 772220 Fax: (0254) 705956

CIRCLE NO. 166

System Science

C & C++ COMPILERS	
Microsoft C 6.0	£245.00
Quick C with Quick Asm	£98.00
TopSpeed C ext	£285.00
Turbo CH Professional	£215.00
Watcom C	£255.00
Watcom C/386	£645.00
Guide lines C++ 2.0	£325.00

C GRAPHICS	
Halo	£225.00
GSS Graphics Dev T'kit	£385.00
GSS XVT Windows	£385.00
Metawindows	£175.00
MS Windows SDK V.3	£325.00
Essential Graphics	£275.00
PCX Programmers Toolkit	£165.00

C DATAFILE	
Ctree source	£245.00
Btrieve Network	£395.00
XQL	£525.00
CodeBase 4	£195.00
db__VISTA III	£545.00

C SCREENS + COMMS	
C Scape, Look & Feel	£315.00
Vermont Views (new ver)	£325.00
Panel Plus II	£275.00
Greenleaf Datawindows	£195.00
Aspen Curses (DOS & OS/2)	£115.00
Aspen Formation	£225.00
C Worthy	£call
Blaise C Asynch Manager	£115.00
Greenleaf Comms	£175.00
Essential Comms	£195.00

PROGRAMMERS TOOLS	
C Programmers T'box	£165.00
Photo Finish (new)	£245.00
MKS Toolkit DOS	£155.00
VsDesigner (case)	£375.00
Brief Editor	£149.00
Epsilon Editor (DOS/OS2)	£145.00
Vq2 Editor (DOS & OS/2)	£125.00
Kedit Editor	£115.00
PVCS Prof	£475.00
PolyMake	£229.00

ASSEMBLERS & LINKERS	
MS Asm 5.1 & Codeview	£95.00
Advantage Disasm	£145.00
Turbo Debugger V.2	£110.00
RTLink Plus	£345.00
Plink 86+	£315.00
Source & Bios (DisAsm)	£125.00

CROSS DEVELOPMENT	
Introl C/Mod-2 68000	£1445.00
Aztec C Cross	£265.00
Avocet AvCase 8051	£795.00
IAR C Cross 8051 etc	£1095.00
2500AD Cross Assemblers	£135.00
IAR Cross Assemblers	£185.00
Simulators & others	£call

FORTRAN	
Lahey Fortran F77L	£395.00
Lahey F77L-EM/32 (386)	£795.00
MS Fortran-77 v.5	£245.00
Salford FTN77/386	£695.00
InGraf for Fortran (source)	£175.00
many Fortran Libs	£call

PASCAL, BASIC	
Turbo Pascal	£75.00
MS Quick Pascal	£65.00
Object Professional	£95.00
Blaise Power Tools Plus	£89.00
Blaise Asynch Manager	£89.00
MS Basic Prof Dev System	£275.00
MS Quick Basic	£65.00
QuickComm	£95.00
Crescent QuickPak Prof	£129.00
Hammerley ProBas	£135.00

OTHER LANGUAGES	
Smalltalk V	£69.00
TopSpeed Modula-2 ver. 2	£145.00
Microsoft Cobol/2	£445.00
PC Logo	£75.00
UR/Forth by LMI	£225.00

MEMORY, DISK & MATHS	
386 Max Prof	£90.00
QEMM/386 5.0	£60.00
QRAM with Manifest	£55.00
Above Disc (LIM Emul)	£69.00
Norton Utilities Advanced	£79.00
SpinRite II	£69.00
Vcache	£49.00
Derive Symbolic Algebra	£115.00
MathCad	£345.00
Statgraphics	£545.00
Mathematica	£call

OS/2, UNIX + 386	
IBM OS/2 SE 1.2	£250.00
MS OS/2 PM Toolkit	£345.00
SCD Unix 386	£call
Interactive 386/ix	£call
LPI Language	£call
VM-386 multi-user	£call

A SAMPLE OF OUR EXTENSIVE RANGE
PLEASE ENQUIRE FOR UNLISTED ITEMS

3-5 Cynthia Street,
London N1 9LF
Fax: 071-837 6411

VISA Access

Prices are subject to change and alteration Prices include UK Delivery

071-833 1022

CIRCLE NO. 167

Xtx

Our UNIX sage, Peter Collinson, is a self confessed ten-thumbed typist with amnesia. However, he has devised ways to overcome his handicaps.

I have always been a dreadful typist. It's not that I don't know where the keys are, or that I look at the keyboard all the time - but somehow I always seem to be hitting the wrong key. Usually it's the one next to the one I want, or a sequence close to the one I had in mind. Luckily the Delete key on my keyboard is a big one.

The disability makes it difficult to type commands into the shell accurately and I do sometimes find myself typing some mis-spelt command again and again. I have resisted the temptation to alias `vay` to `cat` (Figure 1), but have succumbed to a permanent alias of `dc` to `cd` – largely because `dc` is such a pain to get out of when typed accidentally. As a result of all this, I have always invested time into systems which will help me to avoid typing.

Make

One of these systems is `make`. Simply typing

```
make
```

is always much better than attempting the complicated `cc` instruction needed to compile that program.

I tend to insert various targets into my `makefile`'s, like:

```
clean:
    -rm -f $(OBJS) $(TARGET)
    -rm -f core errs
```

The hyphens before the `rm` commands mean 'continue even if these commands fail', otherwise `make` will stop when `rm` fails to find any file to delete. The `$(OBJS)` string is replaced by a list of object files and similarly the `$(TARGET)` string is the main executable being created by the `make`. Both of these are assumed to be declared earlier in the `makefile`. I can now always type:

```
make clean
```

at the shell to ensure that the directory is stripped down to its essential source files.

The ability to have different 'dummy' targets like `clean` means that `make` is a useful way of storing recipes for processing general data. It should not be thought of as solely a programming tool. For example, if you use `troff` for document processing, then a `makefile` can easily be used to contain all that pipeline stuff needed to sequence the various utilities to process the data. Consider this `makefile`:

```
draft:
  tbl dc.mm | nroff -mm | col
top:
  tbl dc.mm | psroff -mm | \
    lpr -PPostScript
```

This makes no clever use of make but means that you simply need to type

```
make
```

to see the draft output, and

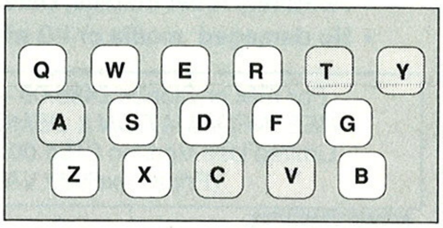
```
make top
```

can be used to deliver the final copy to the printer.

Brain can't take the strain?

`Make` also helps me with another failing: loss of memory. Once a program is written and out of the way, I can never remember it. I immediately forget how it works, what was nice or nasty about it. Memory sometimes comes flooding back when I see the source, but I often have to work out why I inserted those stupid bugs in the first place.

Figure 1
Typing 'vay' instead of 'cat'.



`Make` removes some of the uncertainty. The existence of a `makefile` with its rules for creating the program means that I can return several years later without having to remember what commands were used to create the binary and what particular libraries were included.

However, the bag and baggage associated with the program is increased. In addition to the source and header files, we now have a `makefile`. This can feel like overkill, especially when the source for the program is a single file.

Xtx

Some of my feelings about this surfaced when I came across the simple idea behind the `xtx` program. `Xtx` is a program designed to look inside a file and execute a command that it finds embedded there. The file is not a shell script but source data for another command - perhaps the file contains C source and I want to run the compiler on it to generate an executable program. Mostly these days I use the command in `troff` source files, a comment line in the document will contain something like:

```
<-xtx-*> psroff -ms -t %
```

The string starting with '<' and ending with '>' is a marker that will hopefully never naturally occur in any text file. When I say

```
xtx file.ms
```

`xtx` looks for the marker string, eats any optional white space after the trailing '>', and takes the remainder of the line to be a command passed to the shell for execution.

```
#!/bin/sh
# simple xtx shell script
#
case $# in
1)
  filename=$1
  ;;
*)
  echo 'Usage: xtx filename'
  exit 1
esac
# look for <-xtx-.> using grep
# the dot matches any character
#
# the use sed to remove the marker
# and change % into filename
# The use of '...'"..."
# is intentional
# Type the quote characters carefully
command=`grep '<-xtx-.>' $filename |
  sed -e 's/^.*<-xtx-.> //'"
  s/%/$filename/"`
eval $command
```

Figure 2 - Simple xtx shell script.

Here's what they say about Zortech C++

"*Zortech is a truly fine compiler...If you've been waiting for a major player to offer a professional C++ development system for OS/2 and Windows, as well as DOS, wait no longer... Zortech has it!*"
Richard Hale Shaw, PC Magazine, p.38, March 13, 1990

"' Zortech has done a commendable job with C++ 2.0 and I recommend it highly...The debugger is impressive...Get the Developers version...it's worth the money."
Bruce Eckel, Micro Cornucopia, pp. 8-17, March 1990

" Zortech C++ is one of the best MS-DOS products I've had the luck to use.....I can highly recommend the Zortech 2.0 release."
Scott Robert Ladd, Dr. Dobbs Journal, pp. 64-73, January 1990

"*The Zortech compiler is a well-crafted product; in particular it is efficiently documented and it produces high quality object code. I can recommend it as a fine package*"
Paul G Smith, .EXE Magazine, pp. 16-22, March 1990

AT&T™ C++ V2 Specification
✓ Multiple Inheritance
✓ Type Safe Linkage
✓ Pointers to Members

Compiler Features
✓ Native code compiler with separate global optimzer
✓ Improved MSC Source Level Compatibility
✓ MS Windows™ Compatible
✓ CodeView™ Compatible
✓ Fast Graphics Library with C++ interface
✓ Easy to use TSR functions
✓ Standard Library Source Code included with Developer's Edition
✓ Seamless LIM/EMS Support via new handle pointers or directly via EMS library functions.
✓ Full MS Mouse Library
✓ OS/2 Compiler Option
✓ 99% ANSI C Compatible
✓ Improved code size/speed

C++ Source Level Debugger
✓ Also Debugs C
✓ Assembler Debugging with access to registers and memory.
✓ 16 Debugging Windows
✓ Multiple Statement Lines
✓ Break/Trace/Watchpoints
✓ Dual Monitor Support
✓ Full C++ name unmangling for easy use
✓ Block memory write protect

C++ Tools Classes
✓ 25 C++ Classes with full source code
✓ Includes new Text User Interface Classes
✓ Event Queue, BCD Maths, Linked Lists, Money, DOS error handling classes, text windows and editing classes, virtual arrays, time and date handling, directories and filenames, interupt vectors, etc...

PRICES

C++ Compiler	£129.95
C++ Debugger	£99.95
C++ Tools	£99.95
Library Source	£99.95

Save £130 - Get the Developer's Edition for only £299.95 (includes all the above items).

OS/2 Option	£99.95
C++ Video	£299.95

USA: Zortech Inc.
4-C Gill Street
WOBURN MA01801
Voice: 617-646-6703
Fax: 617-643-7969

EUROPE: Zortech Ltd.
106-108 Powis Street
LONDON SE18 6LU
Voice: 44+ 81-316-7777
Fax: 44+ 81-316-4138

"ANNOUNCING V2.1"
640K Memory Barrier Smashed!

- **New** VCM™ (Virtual Code Manager) technology
- **New** Rational DOS Extender technology for compiling/debugging massive programs
- **New** Virtual C++ Source Level Debugger requires only 4k RAM!
- **New** Remote Debugging via serial port
- **New** Powerful Environment with Browser
- **New** Completely Revised & Expanded C++ Tools
- **New** Improved Compiler Optimization

Zortech VCM™ for DOS
With Zortech's Virtual Code Manager (VCM) you can compile standard MS-DOS applications containing up to 4Mb of code. VCM is a sophisticated virtual memory system that dramatically improves performance over conventional overlay methods. Naturally, our debugger understands VCM too!

Rational™ DOS Extender Technology...
Version 2.1 incorporates this new technology for compiling and debugging really big programs on 286, 386 or 486 based PC's. You can also use V2.1 together with Rational Systems DOS Extender (purchased separately) to produce your own applications which can access memory beyond the 640k DOS limit.

C++ Debugger in 4k RAM!
Zortech's Virtual C++ Source Level Debugger can now locate itself in extended memory on 386 machines. This requires only 4K of conventional RAM!

STOP PRESS - NEWS FLASH
386 Compiler/Debugger Option (using Phar Lapp DOS Extender), UNIX 386 Compiler and OS/2 Debugger all available soon. Also new C++ Classes and Addison Wesley ZTC++ book.

ORDER/UPGRADE HOTLINE 44+81-316-7777

CIRCLE NO. 168

The '%' character provides a convenient way of saying 'here's the place to add the current filename into the command'.

A plausible shell version of the command is to be found in Figure 2 and the full C version in Figure 3.

The C program

The C program expands on the basic idea a little. It's nice to be able to store more than one command in the file and select the one to be used from the command line. This is done by replacing the star in the marker string by another letter and then selecting that letter from the command line. For example, you can run `lint` on the C program by typing

```
xtx -c l xtx.c
```

A few caveats about the C program. As you can see it was written in 1985 and I have resisted the temptation to tart it up for inclusion in .EXE in 1990. So, it's not in ANSI C - and doesn't use `getopt()` to decode arguments. I reason that a tested working program is worth more than one which is messed up for religious or cosmetic reasons. If you're a C programmer - then sorry about the bracketing style. Actually, I'm not that apologetic - I like to have opening '{' and closing '}' lined up in the text. Old habits die hard.

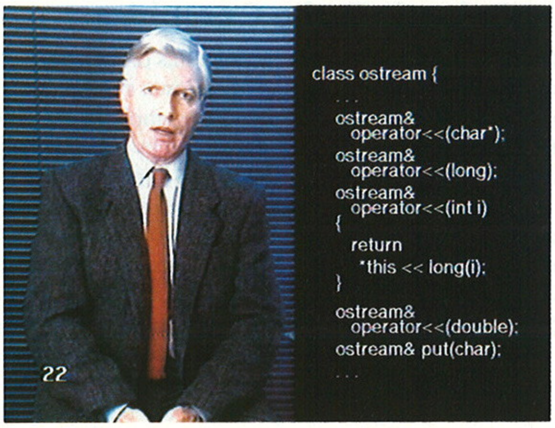
(*Continued on page 70*)

```
/***

* program name:
  xtx.c
* function:
  Looks in a file for a string of the form
    <-xtx-[letter]>
  If the string is followed by a line ending with a \n
  Then takes the line and executes it using /bin/sh
  The letter is used to select one of a number of commands
  which may be embedded in the file. The character '*' is
  used to mean the default.
* switches:
  -c [letter] select a particular line in the file
  -e   echo the matched command from the file
  -a   print all matched commands from the file and their
      key letter
  files
  -    Take data from standard input for scanning
* compile time parameters:
<-xtx-*>  cc -o xtx -O xtx.c
<-xtx-l>  lint -h xtx.c
* history:
  Written October 1985 Peter Collinson
***/
#include <stdio.h>
#include <ctype.h>

char  cmdbuf[BUFSIZ];   /* Where to store the command */
char  expanded[BUFSIZ];  /* expanded version - includes filename */

char  selchar='*';     /* selection character */
int echosw;           /* set if echo only */
int allsw;           /* print all matched commands from the file */

#define WILD  1        /* Wild character in the string below */
char  matchs[] =       /* Used to find a match with a command */
{  '<','-','x','t','x','-',WILD,'>'  };
#define STORESTATE (sizeof matchs)

char Usage[] = "Usage:xtx [-e][-a][-c letter] files..\n";

main(argc, argv)
char **argv;
{
  register char *p;
  FILE *fd;

  while (--argc)
  {  p = *++argv;
    switch (*p)
    {
    case '-':
      p++;
      switch (*p)
      {
      case 'a':
        allsw++;
        echosw++;
        break;
      case 'c':
        if (--argc)
        {  argv++;
          selchar = **argv;
        }
        else
          fatal("No argument to -c given\n");
        break;
      case 'e':
        echosw++;
        break;
      case '\0':
        scanfile(stdin);
        if (cmdbuf[0])
          execute(0, cmdbuf);
        break;
      default:
        fatal(Usage);
      }
      break;
    default:
      if ((fd = fopen(argv[0], "r")) == NULL)
      {  perror(argv[0]);
        break;
      }
      scanfile(argv[0], fd);
      (void) fclose(fd);
      if (cmdbuf[0])
        execute(argv[0], cmdbuf);
      break;
    }
  }
  exit(1);  /* Command has failed */
}

/*
 * Scan a file looking for an appropriate line
 */
scanfile(fname, fd)
register char *fname;
FILE *fd;
{
  register c;
  register state = 0;
  register char *cmdp;

  while ((c = getc(fd)) != EOF)
  {  if (state == STORESTATE)
    {  if (isprint(c) || isspace(c))
      {  if (c == '\n')
        {  *cmdp = '\0';
          if (allsw)
          {  execute(fname, cmdbuf);
            state = 0;
            continue;
          }
          return;
        }
        else
        if (cmdp < &cmdbuf[BUFSIZ])
          *cmdp++ = c;
        continue;
      }
      state = 0;
    }
    else
    if (matchs[state] == WILD)
    {  if (allsw)
        selchar = c;
      if (c == selchar)
        state++;
      else  state = 0;
    }
    else
    if (c == matchs[state])
    {  if (state++ == 0)
      {  cmdp = cmdbuf; *cmdp = '\0';  }
    }
    else  state = 0;
  }
}

/*
 * Call the shell
 */
execute(fname, cmdp)
register char *fname;
register char *cmdp;
{
  register char *p;
  register char *f;

  for (; isspace(*cmdp); cmdp++);

  if (fname)
    for (p = expanded; *cmdp; cmdp++)
    {  if (*cmdp == '%')
        for (f = fname; *f; *p++ = *f++);
      else
        *p++ = *cmdp;
    }
  *p = '\0';

  if (allsw)
    fprintf(stderr, "%c\t%s\n", selchar, expanded);
  else
    fprintf(stderr, "%s\n", expanded);
  if (allsw)
    return;
  if (echosw)
    exit(0);
  execl("/bin/sh", "sh", "-c", expanded, 0);
  perror("Exec");
  exit(1);
}

fatal(str)
char *str;
{
  fprintf(stderr, str);
  exit(1);
}
```

Figure 3 - The xtx program.

VIDEO COURSE

Learn C++



NO RUSH!
NO TRAVEL!
NO HOTELS!
ONLY £299.95

Learn C++ Now!
The great exodus of programmers from C to C++ has begun! Since C++ builds on C, it's the easiest OOP language to learn. That's why it's called "the language of the 90's".

Why the rush?
PRODUCTIVITY! Yes, C++ programmers can write programs in less time requiring less maintenance. Large projects become much easier to manage.

Unfortunately, learning C++ can be very costly. Classroom instruction is expensive even without the travel and hotel costs. Of course, not learning will cost you even more in the long run. Now there is an alternative!

The top C++ video tutorial at the lowest possible price.
The C++ video tutorial from Zortech is the ultimate C++ training tool for work or home at only £299.95. It comes on six VHS video tapes containing 32 clear, extensive tutorials.

Used in conjunction with the concise workbook and tutorial disk, you will find everything you need for fast-track C++ tuition.

The course is generic (i.e. compiler and hardware independent) and is available with or without the award winning Zortech C++ Compiler for MS-DOS and OS/2.

Start writing C++ code within a week.
As a C programmer, you will start producing C++ code within a week of concentrated use of this course. Alternatively, spend just an hour a day watching the video and working through the suggested exercises to learn C++ in only six weeks!

The Leader in video tutorials.
If you don't already know C, you can join tens of thousands of programmers world-wide who have learnt C with the Zortech Complete C Video Course, described as:

"An excellent bargain ...
... I heartily recommend"
Gary Ray, PC WEEK

Save your company thousands of £££'s.
One programmer can train for only £299.95, but you can train ten programmers for just:

C++ Course	£299.95
9 Extra Workbooks	£179.55
Total 10 students	£479.50

Yes! Only £47.95 each! (With all the FREE refresher courses you need!)

Now, Zortech with its new C++ Video Tutorial has refined the art of video tuition and presentation even further for the 1990's.

Only £299.95 complete
- Six Videos with 32 lessons
- 256 page workbook
- Tutorial disk
- Compiler & hardware independent
- NTSC or PAL format
- Tax deductible

Don't delay, order now!
Just mail the coupon or call the order hotline for same day shipment.

USA: Zortech Inc.,
1165 Massachusetts Avenue,
ARLINGTON, MA02174
Voice: 617-646-6703
Fax: 617-643-7969

EUROPE: Zortech Ltd.,
106-108 Powis Street, LONDON
SE18 6LU
Voice: 44- 1-316-7777
Fax: 44- 1-316-4138

IN EUROPE CALL:

44- 1-316-7777

ORDER FORM
Please rush me these items:

Qty	Description	Price
___	C++ Video Tutorial	£299.95
___	Extra C++ Workbooks	£. 19.95
___	Zortech C++ Compiler	£129.95
___	Developer's Edition	£299.95

.EXE 3/90

Please add £5 + VAT postage in UK
All UK orders please add 15% VAT
International shipping charged at cost

Name ____________
Company ____________
Address ____________

Phone ____________
MC,VISA or CHQ ________ Expiry ____
Card No. ____________

CIRCLE NO. 169

Books

The EISA standard for PC architecture is celebrated in a book of its own; and a stylish introduction to Turbo C's graphics.

EISA not neisa

EISA. Yes you do, the Gang of Nine, two fingers waved at IBM, brave new alternative to MCA bus. All that. Now that we are beginning to see some EISA-class machines in the windows of the dealers, what a good opportunity for somebody to bring out a nice little pamphlet, for the benefit of the lay programmer/user who does not intend to run up a network card, this week.

Inside the EISA Computers is a little pamphlet wrapped up in a rather larger book. I challenge the author, Tony Dowden, to put his hand on his heart and tell me that the chapter on MS-DOS V4.0 commands (yes, DIR and that crew) serves any purpose other than to dirty a few blank pages with ink. Similarly, we have a chapter of serial port documentation which contains diagrams of the pin arrangement of both 25 pin *and* 9 pin D-type connectors, suggests how to wire null modem cables, initiates us in the mysteries of serially transmitted data, even throws in a couple of assembly language routines that use INT 14H. Chapters on The Operating System, VGA, the Parallel Printer and 80486 similarly take the Michael.

So what is one left with, once the explanations of Binary Numbering Systems and BIOS calls have been hacked away? Well, there is a diagram of the rather clever way that the EISA connector is designed, so that EISA cards and ISA cards use completely different sets of contacts, although they both fit into the same hole. There is a list of all the signals on the bus, a brief description of the DMA system and an I/O memory map. EISA includes a neat configuration facility, so that a board manufacturer can include data about his cards in the machine's SETUP facility. There is a big chapter about this, with loads of interesting-looking C code; but I can't see it being of any use to anyone except a board manufacturer.

The first book about something new is often disappointing. This is, and it is (if you see what I mean). Let's hang on; there will be another along in a minute.

Inside the EISA Computers — *Author: Tony Dowden*
Publisher: Addison Wesley — *Price: £19.75*
ISBN: 0-201-52397-3 — *Pages: 236*

Quixotic Turbo C graphics

A lot of books come through this office - perk of the job and all that - and I have developed my own system of classification. The volume in question is a good example of the *John Wiley Mildly Eccentric* - the genre characterised by a slightly off-beat approach to a well-trodden topic. *Graphics Programming in Turbo C* is yet another tutorial intended for the beginner - probably one who has only just sorted out his `floats` from his `doubles`. However, if this happens to be your current position, I think that this would be a good choice of text to help you cut your graphical teeth.

The approach is closely tied to Borland's BGI graphics library, so you really will need a copy of Turbo C V2.0 or better. The book is packed with lots of little examples which grow in complexity towards the back. Many of these are beautifully original - at least in my experience. The arc and circle functions, for example, are used to create an exquisite net pattern (as well as the traditional rectangle with rounded corners). The fill routines are pressed into service (in about a page of code) to produce the elegant rendition of Escher's 'impossible' triangle which adorns the cover of the book. The illustrations, incidentally, are all simple monochrome screen dumps; but they have a uniformity of style which makes them more pleasing to look at than many a flash colour plate.

It's not just pretty. A chapter on mouse input is most welcome (since there's no direct support for this in Turbo's library; one is obliged to mess about at interrupt level), as is some practical code for doing menus and pop-ups. I also liked the program which produced HP-GL output, written to a file. The idea is that you can import your Turbo C graphics into WordPerfect V5.0, and so become a desk-top publisher.

Less originally, the *pièce de résistance* is a cut-down CAD application, which is 16 pages of C long. Never mind. A tear out flyer stuck in the front offers the accompanying disk set at £12.65 - which is nearly as much as the book itself.

Graphics Programming in Turbo C — *Author: Leendert Ammeraal*
Publisher: John Wiley — *Price: £12.95*
ISBN: 0-471-92439-3 — *Pages: 199*

UNIX Regular *(Continued from page 68)*

The C program does have some interesting coding aspects. The work gets done in the `scanfile()` routine, and this is coded in an attempt to make it scan data as fast as possible. I avoid examining the data twice by NOT calling `fgets()` or gets`()` to get a line of data and then scanning that data for the magic string.

It's much faster to use `getc` to get a character, then have a finite state machine to recognise the string that we are looking for. The main loop of `scanfile()` is a call to `getc` and a set of tests. `getc` is implemented as a macro, generally accessing the read buffer directly and so is fast. The `state` variable is used to select the exact comparison to be made for the particular character that has just been read. If a successful test is made, the state is moved to the next one; and the next character in the `matchs[]` vector is sought in the input. The remainder of the input line is stored if a successful complete match is found.

Acknowledgements

This idea was originated by Tom Duff from AT&T Bell Laboratories. His command is called `com`, because it is for COMpiling files. I had the idea from David Tibrook.

In the absence of source, I took the idea and created `xtx` named because you are eXecuting from TeXt. The command is pronounced 'ex-tex'. Enjoy.

EXE

Peter Collinson is a freelance consultant specialising in UNIX. He can be reached electronically as `pc@hillside.co.uk` *(although your mailer might be happier to put the address the other way round) or by phone on 0227 761824.*

Training for developers, by developers

Clipper 5.0

The Professional Developers Course

Our courses are geared towards applications development with emphasis on the use of structured programming to design, build and maintain libraries of reusable system building blocks.

We are a Nantucket Authorised Training and Development Centre.

Systems Development — CASE
Consultancy — Structured Methods

Applications Technology

Training — Clipper, C, ASM, SQL
Project Management — Networks, Windows

IT's easy with APT

James Ormrod
Applications Technology Ltd
27 Park View, Crowmarsh
Wallingford, Oxon. OX10 8BL
Fax 0491 32002 ☎ 0491 35834

CIRCLE NO. 170

Before you leap™
Software for Software Estimation
using Function Point Analysis & COCOMO

The leading Software Estimating Tool for the 90's

NOW... with a 'Big Brother' MK II version runninging under Microsoft's Windows™

ORIGINAL	MKII
METHODOLOGY FOR SIZING	
Function Point Analysis (IBM	MKII Function Point Analysis (Nolan Norton)
Data Diagram	SSADM/Entity Diagrams
(Inputs/Outputs/Files)	(Transaction Analysis)
METHODOLOGY FOR ESTIMATING	
COCOMO	COCOMO (Enhanced Version)
	FPA MK II Method (Nolan Norton)
OPERATING REQUIREMENTS	
PC/XT/AT/PS2 & Compatibles	PC/AT286 & 386-Compatibles
MS/PC DOS 2.1x	Microsoft Windows
Single User Version	Single & Network Versions

KEY FEATURES

Automatic Recalibration - 'What If' Evaluations - Life Cycle Projections
Demonstrations Accuracy - Modifiable Cost Drivers - Data Export Facilities
Full Colour Graphics - Highly Interactive - Access to multiple projects at any time

Software overruns are a company's nightmare that can be avoided by careful estimation based on **facts,** not guesses. ***Before You Leap*** provides a knowledge based tool that continually calibrates to YOUR ACTUAL EXPERIENCE.

Wouldn't that be a better way?

Call us now for further details and an evaluation system.

Strategic Systems Technology Limited
14 Landons Close, Prestons Road, London E14 9QQ
Tel:- (071) 538 8228 Fax:- (071) 515 3887

CIRCLE NO. 171

System Science

Microsoft Languages

C Professional Development

The latest C 6.0 has improved optimisation, in-line assembler and draft ANSI support amongst other enhancements. The new Programmer's Workbench integrates the operation of the compiler, make, linker code browser and the new Microsoft C Advisor. CodeView 3.0 can reside in extended memory for DOS. Dos and OS/2 support is included for just........... **£245.00**

Basic Professional Development

New version 7.0 handles large applications with run-time overlay support, far strings and EMS4.0 support. New included toolboxes for menus and user interfaces, presentation graphics and maths. New ISAM for transaction processing and a data dictionary. The Microsoft Basic advisor is in the Extended Quick Basic environment. DOS and OS/2 ...**£275.00**

Windows Software Dev. Kit 3.0

Windows SDK provides a set of tools and documentation for developing applications for the new Windows 3.0, including a resource compiler, dialog editor and libraries for DDE (dynamic data exchange). Use it with Microsoft C 6.0 or 5.1 to put the most into your Windows app. **£325.00**

Windows 3.0 .... **£89.00**

Fortran-77 Compiler ver 5.0

Fortran 77 with VAX and proposed 8X extensions, the compiler supports large arrays, large data and large code and graphics primitives. Editor, linker and CodeView, the source code debugger, are included. For DOS and OS/2 .....**£245.00**

OS/2 PM Toolkit

For OS/2 ver 1.2, the PM toolkit contains extensive documentation on OS/2 and PM programmming, with software tools, such as the Icon Editor, Dialog Box Editor and resource compiler, with the on-line Quick Help system. Use the toolkit with C, Fortran or Basic **£325.00**

Further Microsoft Products

Assembler 5.1 (DOS and OS/2)......................£95.00
Quick C ver 2.5 ....£69.00
Quick C with Assembler ver 2.51.......................£98.00
Quick Basic ver 4.5 ....£69.00
QuickPascal ver 1.0 ....£65.00
MS Cobol ver 3.0 (Dos and OS/2) £475.00
Microsoft Programmers Library (CD-ROM)...£275.00

3-5 Cynthia Street,
London N1-9JF
Fax: 071 837 6411

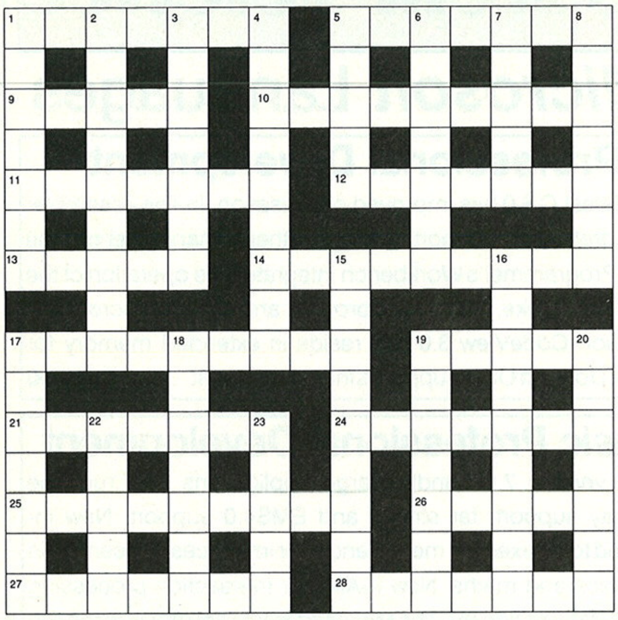
Access VISA

Prices are exclusive of VAT, but do include UK delivery.
Prices are subject to change and alteration.

071- 833 1022

CIRCLE NO. 172

.EXEWORD



ACROSS

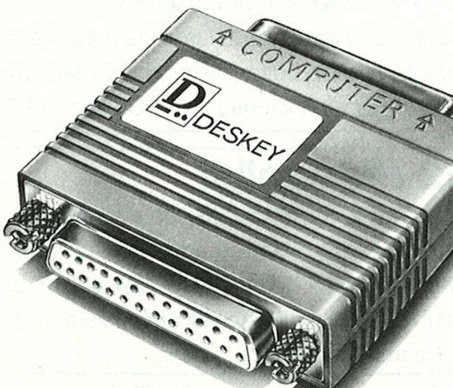
1 Micro output collection may translate (7)
5 A sugary system (7)
9 Earful on the line (5)
10 Surgical system? (9)
11 Fighting medievally, headlessly pushing out (7)
12 Three-toothed missile (7)
13 Wear away as Elizabeth went by horse (5)
14 Best left to experts with technique? (9)
17 Before screen flasher that gives warning (9)
19 Pentavalent impurity with blood to spare? (5)
21 Summary to be coded later (7)
24 Brings in current data (7)
25 Get rid of broken line I tame (9)
26 Lager ones best avoided (5)
27 Wandered off with food carrier in backward ministry (7)
28 Spanish festivals for screen animations (7)

DOWN

1 Cheer up the operator's desk (7)
2 It gives immediate access to data (4,5)
3 Tick off and list chunks of data (7)
4 Increases size of almost lone logic elements (9)
5 Creative person's job finder (5)
6 Where you get to travel on a network (7)
7 Aries may somehow lift up (5)
8 Two-fingered in our cases (7)
15 Cyclic displays of screen pages (9)
16 Build an algorithmic base (9)
17 Business between input and output (7)
18 Useful little routine (7)
19 Cheat on the roof with a fresh start (7)
20 How chips' coating opposes current (7)
22 Three way start then hesitation for game one (5)
23 Antelope from oriental country (5)

C	O	M	P	U	T	E	R	I	S	A	T	I	O	N
H		O		L		X		O		P		T		E
A	N	A	L	Y	S	T		N	O	T	H	I	N	G
N		N		S		R		I		N		N		A
N	E	S	T	S		I	N	C	R	E	M	E	N	T
E				E		N				S		R		E
L	I	A	I	S	E	S		D	I	S	B	A	R	S
		B				I		U				R		
T	R	A	F	F	I	C		M	I	S	T	Y	P	E
R		N		I				P		T				A
I	N	D	U	C	T	O	R	S		O	U	T	E	R
G		O		T		M		R		L		I		M
G	E	N	U	I	N	E		I	N	E	R	T	I	A
E		E		O		G		S		I		L		R
R	E	D	U	N	D	A	N	C	Y	C	H	E	C	K

.EXEWORD JULY



DESkey
THE COMPLETE DONGLE RANGE

Data Encryption Systems Limited have more experience in the design of software protection devices (dongles) than any other company in the U.K. *12 years* experience in the design of dongles have gone into our latest 2 products: The DK1 and The DK12, both of which use the very latest in ASIC technology.

- 128 BYTES of memory split into 2 x 64. *Half is intended for both read and write, the other half is read-only. Writing to this protected half can only be performed using your unique password. This feature can be used for tasks such as identifying the modules of a multi-module package which have been paid for.*
- Down Counter. *This enables a customer to be sold 'n' goes of the software. After which the DK1 stops working until the counter is reset using your password.*
- Pseudo Random Number Generator. *Billions and billions of bits without repeating! Software and Data Encryption could not be easier. 'Seedable' too.*
- Completely Transparent to Printers, End Users and anything else requiring the port. *16 or more on any one parallel port.*
- No Hidden Extras. *Everything you need is in the DK1. No need for extra 'programming' units.*
- DESlock Software *uses DK1 or DK12 to provide instant encryption of .EXE or .COM software without the need for access to source (or .OBJ). Will even protect DOS commands. Keeps you/your employees the right side of the copyright laws.*
- Designed and manufactured by the U.K.'s leading software security specialists *Data Encryption Systems Limited.*
- DK12 *is a reduced feature DK1, available at a quantity price to beat ALL the competition (ask for a quote).*
- Serial and Bus versions available.

For further information, contact us at,

Data Encryption Systems Ltd

Edbrook House, Cannington, Bridgwater, Somerset. TA5 2QE
Telephone (0278) 653456 Fax (0278) 653300

CIRCLE NO. 173

ANOTHER LOUSY AD!

Our Ads get lousier and our software keeps getting better! Because we'd still rather spend time perfecting our software than writing Ad copy!

A lot of **Clipper** developers must agree with us, because readers of Data Based Advisor Magazine recently chose **NetLib** as the most popular Clipper add-on in the LAN connectivity category. And since 1985, it's been one of the most popular in *any* category!

Why? The answer is simple. Getting **NetLib** is like getting many libraries in one. Here are just some of the features you'll see:

- So many new locking functions, you'll wonder how you got by with just Rlock and Flock!
- Timer functions to allow you to cancel a Read and unlock a record automatically!
- Station-to-station communications under complete program control.
- Ability to increase the number of file handles, *and* many dBase file functions like DBF() and NDX() that Clipper left out.
- Extensive NetWare print control including the ability to select print queues by name.
- Automatic, dynamic encryption with over 10^{19} keys, giving the best possible security short of special hardware.
- Automatic journaling for audit and history reports.
- Includes Rettig *and* Norton Help.

Data Based Advisor Magazine said " Without a doubt, **NetLib** gives the greatest amount of flexibility in writing applications for the Novell environment."

We are so sure that you'll find **NetLib** indispensable that we're one of the few companies to offer a **30 day unconditional guaranty**. You *must* be satisfied!

Order now and you'll get a free NetLib update for Clipper 5.0, when available.

Helping Clipper programmers out of tight corners since 1985. SM

1989 DATA BASED ADVISOR The Database Management Systems Magazine Readers Choice Award

Communication horizons

RaSQL, Clipper, NetWare SQL: A Sequel Without Equal. Would you believe it if we told you that you can write **multi-user, SQL applications** with *Clipper* today? That you can use your existing hardware? That you can potentially boost performance with database engine technology? That you can begin writing applications within a day?

Prepare to become a believer. The company that has been the Novell connection for Clipper since 1985 now brings you **RaSQL**, the first library to interface Clipper with Novell's NetWare SQL. Here are some of the things RaSQL allows you to do:

- Write programs with Clipper-87, or with the new Clipper version 5.0. We give you both libraries!
- Access data using SQL, or our easy DBF-like access commands (Use, Seek, Set Order, etc.)
- Open multiple tables/views concurrently.
- Access DBFs and SQL tables in the same application.
- Download data from a table to a DBF and vice versa.
- Get pop-up assistance with Rettig or Norton Help. We've included both data files on the disk!

RaSQL uses NetWare SQL, the *only* database engine that runs right on the server, requiring no additional hardware or installation skills. You can even access existing **Btrieve** files, including those used by **Platinum**.

We are so sure that you'll love **RaSQL** that we are one of the few companies to offer a **30 day unconditional guaranty**! If you think RaSQL is not fitting your needs, we'll take it back and quickly refund your money. You *must* be satisfied!

Order now and you'll get a free RaSQL update for Clipper 5.0, when available. Ask about our special price for NetWare SQL, too!

If you've ever wanted to take the database server plunge but have been afraid to because of the commitment in time, equipment, cost and training, **RaSQL** is your answer. You'll soon be writing applications that will have your users and associates saying "How did you do that, you RaSQL?"

701 Seventh Avenue • Suite 900
New York • New York • 10036
Tel: 212-724-0150 • CIS: 75170,1032
Fax: 212-840-1555 • MCI: 367-1186
Available in the UK from: Software Paradise: (0222)887521

CIRCLE NO. 174

GERMANY BECOMES THE TARGET AS EUROPE BOOMS

Recruitment experts are keeping their eyes on the economic and political changes in Germany in the hope that unification will open up the contractors market.

Many agencies have traditionally relied on the Benelux countries and France to hire UK staff, while Germany has been regarded as a relatively closed market. Observers think, however, that the barriers will soon come down.

Peter Jobson, regional manager at Datapool Consultants, said that a large market was likely to emerge as restrictions are eased. He maintains that foreign contractors have had difficulties in the past because trade union agreements were favourable to local staff, and he cites a recent case where UK contractors were physically ejected from a firm by union members.

Paul Smith at Real-Time Consultants also complained that German legislation is 'far too restrictive' at the moment, but said one of the next tasks for the company is to look at operating there.

Most agencies operating in Europe are optimistic about the current state of the market, which has not suffered from the downturn seen in the UK this year. Datapool's Peter Jobson said: 'The market is very good indeed. The rates are very reasonable, taking into account the cost of living in other countries. It's all very attractive for the British.'

David Mitchell, overseas director at Hunterskil, added: 'In the UK, firms are backing off doing major projects if they can. But it's pretty much the status quo over there. If anything they're probably more geared to 1992, and perhaps they're developing more than usual, particularly centralised operations like government bodies.'

Traditionally, the most popular countries for UK freelances have also been the nearest, and contracting in the Benelux states has been possible almost as long as it has in the UK.

The tax and national insurance regulations are complex and vary hugely from country to country, however. According to agencies, they are notoriously stringent in the Netherlands.

Hunterskil's Mitchell said his firm has been involved in negotiations with Dutch tax authorities to establish a system which would allow UK contractors to operate there through the agency. The firm has had to set up what amounts to a PAYE scheme for its contractors, where tax is deducted at source with a limited amount of earnings held back to cover expenses.

Paul Smith at Real-Time Consultants, however, thinks the severe tax regime in the Netherlands could put some contractors off: 'I think freelances prefer to operate through their own limited companies and make their own arrangements - part of that includes the flexibility to manage their own financial affairs.'

Real-Time Consultants also deals mainly with Benelux states and France. According to Smith, France in particular has been opening up over the last year, with more opportunities in larger companies.

Keith Rodgers

EXE

DOWNTURN LEAVES CONTRACTORS TWIDDLING THEIR THUMBS

Large numbers of contractors are expected to be out of a job during the rest of this year as the market feels the squeeze and temporary work tails off.

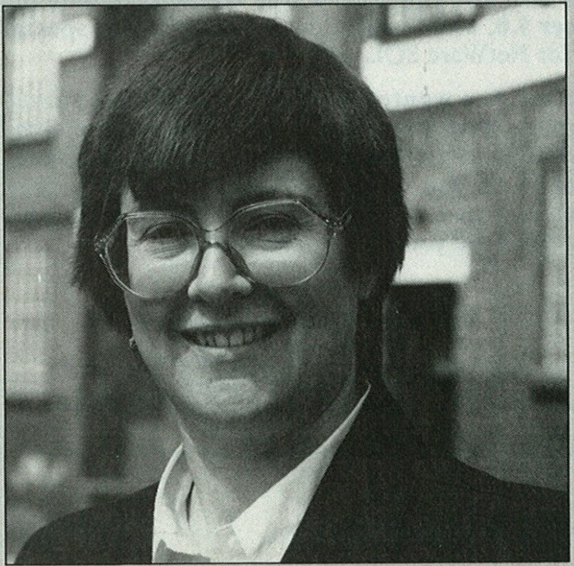
According to the Computer Freelancers Association, even members working in the traditionally strong PC sector are feeling the pinch, as unemployed contractors from other fields move around to find work. The picture is not expected to improve before the end of the year.

Douglas Milnes at the Association said, 'There are a lot of contractors sitting at home and twiddling their thumbs'; including senior systems analysts as well as junior programmers.

The root of the problem is the unsettled economic climate, which has hit the confidence of the market. On top of that, jobs in the City, one of the largest employers of freelances, have been cut. While the Association expects that part of the market to settle, it is unlikely to do so this year.

Milnes also pointed out that a fall in hardware sales last year, which led to a drop in software requirements, was now being felt by contractors. Hardware sales are picking up again, but it could be another twelve months before levels of work follow suit.

Janet Knott, manager of the contracts division at BIS, described the market as 'patchy', and added: 'What we're finding is that people are putting off taking on contractors, but when they want them, they want them yesterday. It's very much a case that, unless people are immediately available, they won't be considered for interview.'



Knott: 'Contractors market patchy'.

Referring to the PC market, she said: 'We're still suffering from the fact that years ago a lot of people set up as consultants working from home. We're finding that, for the last year or so, those people need help when they can't find work, so if anything there's an oversupply in the market.'

For all that, Milnes expects contractors' rates to stay fairly steady because a lot of professionals, used to earning high salaries, won't want to discount. As the market is squeezed, however, they are likely to be more willing to travel further.

.EXE APPOINTMENTS

Recruitment Sales Executive
Jeremy Cox 081-994 6477 x2337

Process Communications Ltd, 10 Barley Mow Passage, Chiswick, London W4 4PH Fax No: 081-994 1533

Rates: 1/4 page	132mm x 88mm	£520
1/2 page	132mm x 179mm	£955
1 page	273mm x 179mm	£1795

Copy Deadline: 15th of the month of publication.

Cancellation: Advertisements may not be cancelled after the copy deadline.

C & PROLOG SPECIALISTS

Quintec Systems Ltd, an expanding and innovative software house specialising in PROLOG-based tools, now has exciting opportunities in Oxford for:

QUINTEC

PROLOG APPLICATIONS CONSULTANCY

To work on consultancy projects and applications/product development in business and technical areas. Commercial experience in using PROLOG, expert systems tools and object-orientated techniques is essential.

DEVELOPMENT/SUPPORT PROGRAMMERS

To work on development and enhancement of our product range and also to provide customer support. At least one year of commercial experience using C at a systems level and some knowledge of PROLOG is required. Knowledge of X-Windows on UNIX, MS-WINDOWS, VAX/VMS and DOS is also desirable.

PROLOG IMPLEMENTATION PROGRAMMERS

To work on further implementations of our PROLOG compiler/environment on SPARC, MIPS and other RISC processors. Extensive experience in C systems programming and assembler in a UNIX environment is essential.

Excellent salaries are offered to the right candidates. Please apply in writing with C.V. to :-

Tim Irving, Quintec Systems Ltd,
Midland House, West Way,
OXFORD OX2 0PL

Tel. 0865 791565

ACTIS

PEOPLE IN UNIX*

LET ACTIS HELP YOU FIND THE WAY

4GLs/RDBMS
COBOL/DIBOL
C Programming
CASE/SSADM

Are you seeking a position in a UNIX environment?

Our consultants will be pleased to help you. We have a large number of instructions from a wide variety of clients in this growing market. A career move is an important step in life and needs serious consideration, we will discuss the opportunities available. If you have some experience and relevant skills and are considering a change of position come and tell us where you want to go.

Send your CV or telephone for an appointment, we will be visiting the following towns during the next three months: **LEEDS, BRADFORD, NOTTINGHAM, BIRMINGHAM, NEWCASTLE, DURHAM, MANCHESTER, SHEFFIELD, LEICESTER, COVENTRY, DERBY, WARRINGTON, LIVERPOOL, CHESTER.**

Consultants
Management
Support
Training

*Unix is a trademark of AT&T

To discuss opportunities or assess your career options on a confidential basis talk to Honor Lindsey **on (0204) 20200 or SEND YOUR CV to ACTIS RECRUITMENT, 17 CHORLEY NEW RD, BOLTON BL1 4QR**

(8306)

ACTIS RECRUITMENT

SOFTWARE PROFESSIONALS

REAL TIME UNIX/C
Surrey/Hants **c£15k to £20k**

My client a small and expanding designer of powerful systems for the Commercial electronics industry seek additional members to join their young design team.

You should have at least 1 years experience of C or Pascal (cross train to C) gained within a realtime UNIX or Vax environment.

CONSULTANCY
Surrey **to £20k**

This small, dynamic systems consultancy seek young software professionals to take responsibility for new projects involving a wide variety of applications tools and methods.

You should have at least 3 years' experience of Real Time software design and the ability to deal effectively at all levels. Projects involve Graphics, Control and Communications. C, UNIX and embedded microprocessor systems design.

Call James Hunt NOW for details. TEL: (0425) 475480

ASH associates

RECRUITMENT CONSULTANTS
3 Pipers Ash, Ringwood, Hants., BH24 1UF
Tel: (0425) 475480 Fax: (0425) 480807

WORK WISE

Workwise is a very successful technical agency, we have placed many software engineers in exciting positions throughout the UK. We urgently have requirements for the following:

SOFTWARE ENGINEERS (MIDLANDS)

Must have Real-Time control experience and a high level language eg ADA, FORTRAN, PASCAL 'C' or ASSEMBLER. Degree or HND and 2 years experience, preferably in the avionics industry.Salary negotiable.

SOFTWARE ENGINEERS (SURREY)

Must have 18 months to 2 years experience of 'C' programming using MS-DOS. ASSEMBLER would also be useful. Working in small design groups. Salaries negotiable

SOFTWARE ENGINEER (ref TV604)

ASSEMBLER exp on various microprocessors, 'C' programming, BASIC (especially HP BASIC) familiarity with PC AT, PS2. Experience with application packages such as spreadsheet, DATABASE (PARADOX) Salary: £NEG.

SOFTWARE ENGINEER (ref TV603)

To develop user friendly software for engineering tools. Experience, knowledge of current telecomms and cellular tools, UNIX, 'C', C+, PASCAL, BASIC, DOS, MICROSOFT, (min of 2 of these). Also hardware knowledge. Degree in computer science with training ability. Excellent prospects. Salary: £NEG + BENEFITS.

If you are interested in any of the above or seek a career change, call Drew or Philip on 071-495 2442
CV's to: Workwise Technical Recruitment
34-36 Maddox Street
London W1R 9PD

Analyst/Programmer **to £20k + benefits**

Esrablished & prestigiously located Human Resources Management Consultancy requires Analyst/Programmer to work on Networked PC's and AS/400. Systems design and integration experience is essential, as is relational database and some 4GL knowledge. AS/400 experience, while preferred, is not essential, and this is an ideal opportunity to expand such skills.

COBOL Programmers - Any Hardware
to £21k + benefits

Major Dealer and Software House requires COBOL prgrammers for retraining to Ingres. UNIX and/or some 4GL experience would be preferred, although commercial COBOL programming is the only essential as full training will be provided.

RPG III/COBOL Programmers - AS/400 **to £26k**

City-based financial information company requires AS/400 Analyst/Programmers, preferably with RPG III on AS/400, although cross-training will be provided from System 38. Also of interest is COBOL programming experience on S3X or AS/400. In all positions a minimum of 18 months commercial programming experince is required, possibly within a financial environment.

C Programmers **to £21k + benefits**

```
INCLUDE OPPORTUNITIES.h /* THE BEST AROUND */
INCLUDE EXPERIENCE.h /* 2 YRS+ REQUIRFED */
DEFINE ENVIRONMENT "Development"
DEFINE PACKAGE "Excellent"
main ()

printf ("call now or send cv for immediate interview/n")
printf ("we meet all relevant candidates/n evenings also/n")
```

Software Engineers **to £30k + benefits**

Major Hardware Manufacturers require the services of 3 Software Engineerstio pioneer a new project under UNIX. The ideal candidate will have a minimum of 5 years experience of COBOL and/or 'C'. The position will be based in the Thames Valley, within easy reach of London. Experience with UNIX is desireable but not essential and a knowledge of ORACLE V.6 would be an added advantage. Prospects within the company are excellent offering both promotion and training with a comprehensive benefits package.

Product Support/Applications Engineer
to £16k + benefits

Manufacturers of handheld PC's in Buckinghamshire, requires a Product Support/Applications Engineer with an in depth knowledge of PC/MS DOS, together with Programming expertise in 'C' and/or PASCAL. Knowledge of UNIX OS/2 assembler and Support/Training would be an advantage.

Systems Software Professional **to £18k + benefits**

The above company also requirres a Systems Software Professional to take responsibility, including supervision of a small team, of a new project. The ideal candidate will currently be using OSI layered architecture or SDLC, and have a minimum of 18 months experience wityh 'C' and 8086 assembler. A knowledge of any of the following, ORACLE, PARADOX, dBase III is desireable. Experience with device drivers (UNIX or DOS), X-Windows, Z80/68000 ICE will be a distinct advantage.

For details of these and other positions please phone Terry Nelson on the number below or out of office hours on 02406 5892

RONICOM
RECRUITMENT

46-47 Pall Mall, London SW1Y 5JG
Telephone: 071-321 0245 Fax: 071-839 7629

SOFTWARE DEVELOPERS

ESSEX **£15K - £20K**
Communications Developer - seeking software engineer with **'C'/UNIX** and any of: **Windows, TCP/IP, VAX, M68000, X.25** or **RDBMS**

DORSET **to £18K**
Developing **Data Aquisition, Display** and **User Interface Design** you should have experience in one of the following: **'C', Macro, Fortran, VAX, VDP, Intel, PLM** or **IBM PC's.**

WIMBLEDON, LONDON **£15K - £25K**
Networked Document Imaging Systems - you should have experience of 'C' under either **MS-DOS** or **UNIX** combined with Networking, **Ethernet** or **Token Ring.**

SURREY **to £21K**
EPOS Development - **'C'** experience required with at least one of the following: **MS-DOS, UNIX, XENIX** or **ORACLE.**

WORCS **£17K - £20K**
Systems Administrator & Analyst/Programmer - any two of the following is needed: **UNIX, MS-DOS, 4GL, 'C', Seachange** or **Chameleon.**

W. YORKS & SURREY **to £17K**
CADCAM Developer seeking programmer/analysts with **'C'** and **MS-DOS** experience. Any Cadcam experience useful but not essential.

SUFFOLK **£20K +**
IKBS - developing knowledge based systems and neural networks. You should have experience in one of: **AI/IKBS, Object Orientated Programming, User Interface Design** or **Relational Databases.**

HANTS **to £20K**
MOBILE COMMUNICATIONS - developing real-time embedded software systems, supported by CASE tools. 2 years plus experience required in **'C', Pascal, PL/M, X.25, Yourdon or De Macro.**

HERTS **£17K - £20K**
DECISION SUPPORT S/W - developing front end dealing room systems for the worlds financial organisations: **UNIX** and **'C'** essential. Any Windows or SUN workstation experience useful.

W.MIDLANDS **£15K - £20K**
DESK-TOP PUBLISHING - exciting state-of-the-art developments to become involved in if you have solid **'C'** and **PC** experience. Any of the following useful: **OS/2, Windows, Presentation Manager, UNIX, DOS, C++** or **RDBMS.**

MIDDX **to £20K**
PC DESIGNER - Exciting **Media Research** establishment is looking for solid **'C'** under **MS-DOS** with any of the following which would be useful: **C++, i80386, OOP, MS-Windows** or **Graphics. FULL TRAINING.**

SURREY **£15K - £23K**
HOLLYWOOD.....develop systems for the film, video and entertainments industry if you have **'C'** in a real-time environment. Any knowledge of the following would be a bonus: **SUN, AppleMac, IBM PC's, UNIX, DOS, OS/9, CD-RTOS,** or **OOD.**

(SAMPLE OF MANY OTHER SUPERB JOB OPPORTUNITIES:)

HERTS -	**Office Automation Development:** £16k - £24k	'C', UNIX, Database Graphics or Comms.
CAMBS -	**Advanced Image Processing:** £14k - £19k	OCCAM, 'C', Parallel C UNIX or Device Drivers.
HERTS -	**Medical Imaging:** to £18k	Real-time, 'C', UNIX or RMX.
LONDON/ - MIDDX	**UNIX Support & Development:** to £20k	Device Drivers, Kerral, Parallel Pr. or OCCAM.
AVON -	**Software Development:** to £16k	Real-time, 'C', firmware i8086, i80186
ITALY £18k - £21k	**Spacecraft Control:**	UNIX, 'C', C++, ADA or OOD

MANY MORE INTERESTING POSITIONS THROUGHOUT THE U.K!
Contact us to find out more about these posts or the many others that we have. As specialists in the industry, we can provide you with a professional and confidential service in assisting you with the right career move. Telephone or write to one of our consultants.

081-392 1514
ACUMEN SEARCH & SELECTION INTERNATIONAL
No 7, Tideway Yard, Mortlake High Street, London SW14 8SN
Tel: 081 392 1514 Fax 081 392 1518

Evenings till 9pm 081-878 8206

ACUMEN
SEARCH & SELECTION INTERNATIONAL

Unix support

- ★ **£18,000-£28,000 + Quality Car**
- ★ **Home Counties**
- ★ **Leading Computer Manufacturer**

Our Client is **still** achieving exceptional success and growth gained from their reputation for providing outstanding customer support and technical excellence in both hardware and software products.

Due to a broadening market strategy and a revenue target of $450 million for 1990, they urgently seek Unix support professionals. You will have a background, preferably within a manufacturer, in technical or sales support with a fascination for resolving in depth customer problems and providing solutions both in house and in the field.

Your expertise will have been gained in one or more of the following areas:-

Unix—Berkeley or AT+T Versions
Sun, Apollo, Tandem, Stratus, Pyramid, HP
Networking/Comms, Graphics, RDBMS, O/Automation

You will receive top class company benefits and in depth product training where appropriate.

If you are looking for rapid career advancement in an open and encouraging environment call **LORRAINE JONES** now on **0483 506610** or send your C.V. today.

BWRS International Ltd.
Woodbridge Chambers, Woodbridge Road, Guildford, Surrey. GU1 4QD

SOFTWARE ENGINEERS ARE YOU DEVELOPING?

Are you a good software engineer in a not-so-good company?? Are your talents being squandered on telephone hot-line support or on mind-deadening maintenance?? If you feel you could be doing better, consider these 100% **Software Development** opportunities.

Software Tools development, to £22000, Bucks/Oxon. Developing an Object-oriented, UNIX-based graphical user interface!! Upgrade your 'C' skills to C++!! Train in X-Windows. Young Company!!!

CASE Tools development, to £23000 plus Car, Essex/Herts. Experienced in developing relational database applications in 'C'?? Learn the Object-oriented approach!! UNIX application-level skills essential!!

Compiler Software development, to £25000, Beds/Bucks. Fluent in 'C'?? Used UNIX kernel tools?? This young UK company with a U.S. subsidiary builds programming tools for the top computer vendors!!

FORTRAN and VAX skills?? £16000 to £18000, Beds/Bucks. Develop bespoke interfaces to real-time systems!! Learn 'C' and database design!! Customer site visits!!

Interested?? For more details contact **ALAN CARNELL** at Concurrent Appointments or mail him your CV in confidence. Informal initial interviews can be arranged out-of-hours and locally to you.

Concurrent Appointments
Software Recruitment

27 FIELD CLOSE
HARPENDEN
HERTS
Tel: 0582 712976
Fax: 0582 764858

AES

TECHNICAL AND EXECUTIVE CONSULTANTS
30 New Street, Worcester WR1 2DP
Tel: (0905) 23444 Fax: (0905) 23393

SOFTWARE ENGINEER up to £18,000 (Gloucestershire)

This is an exciting opportunity for a high calibre Software Engineer to join an international company with a world wide reputation for providing high technology engineering and computer services to the oil industry.

This key position will suit either a recent graduate or a young engineer with two or three years post graduate industry experience. A background in at least one of the following is essential although training will be given where necessary.

* Embedded Systems Development
* Intel Microprocessors/Microcontrollers
* Real-Time Software Applications
* 'C', PL/M, Assembler
* IBM / PC / XT / AT
* Structured Design Methodologies

Equally important is that the ideal candidate numbers ambition, enthusiasm, determination and flexibility amongst their qualities, as well as good communication skills since interaction between departments and clients is a crucial part of this most important position. Opportunities will arise for both national and international travel.

The job package is as you would expect from a market leader and includes profit related bonus, BUPA membership and life assurance. A relocation package is available where appropriate.

Interested? Contact the Software Department of AES at the above Address.

WHERE INTEGRITY STILL COUNTS

The Computer Resources Group specialises in the recruitment of PC professionals at all levels. We have a large number of vacancies in different areas of commercial computing ranging from Project Managers to Junior Support positions.

Please contact us if you have experience in any one of the following areas:-

- *C Analyst/Programming*
- *Fortran/Assembler 8086*
- *4GI Analyst/Programming*
- *Systems programming in C or Pascal under Unix or OS/2*
- *Unix - system support/installation*

We offer a personal and professional service, so whatever your career objectives, please contact us.

Telephone us on
071-494-2041
or fax your details to
071-734 2556.

CRG

Technical Software Vacancies

WINDOWS/OS2/ PRESENTATION MANAGER

London/Thames Valley £18-28K
A joint venture company between a world leading Information Technology centre and various financial institutions is aiming to increase its workforce by a massive 30%. In order to be part of this exciting development you will be degree qualified, have 2 years experience in structured software design and ideally be versed in communications protocol and interface design. In return for your knowledge of the 'C' language, together with MS WINDOWS or PRESENTATION MANAGER you will be actively involved across the entire software life cycle and be offered an attractive salary, bonus scheme, family medical plan, pension scheme and free foreign language tuition! Ref: MJ25100

RECENT GRADUATES M.Sc GRADUATES

Various Locations to £15K
We have a large number of clients ranging from a major international defence contractors through to small, independent software houses who have an urgent requirement for technically qualified graduates with software experience. If you can offer C/UNIX, ADA, PASCAL, COBOL or any Real Time experience we can offer you a number of interesting opportunities. MJ2557

C → C++

N.H. Counties to £30K
Would you like to develop your expertise in Graphics, Database, Data Comms, Object Orientated Design, Expert Systems or Distributed Processing? Would you welcome the opportunity to TRAIN in C++? If so, and you can offer a good technical degree and a minimum of 1 years experience in C/UNIX you could be working on a variety of technically challenging projects within a small, rapidly growing Systems House. Along with this golden training opportunity my client offers private medical care, company pension, bonus scheme and employee share options. MJ2563

NETWORKING/COMMS SOFTWARE

London/H. Counties to £30K + Car
__SKILLS__
· C/UNIX · VAX/VMS · X25 · X400
· PC-DOS · IBM · OSI · SNA
__OPPORTUNITIES__
Consultancy, Finance/Retail/Comms, Design/Development, Commerce/Government
__BENEFITS__
Excellent Salaries, Company Pension, BUPA, private and medical insurance, generous relocation package, varied, interesting projects and superb career development. MJ2570

SECURE COMPUTING

Hampshire to £30K + Car
Our client, a joint venture company between one of the UK's leading technical consultancies and the UK's foremost communications company is actively seeking experienced software engineers to lead the way in Secure Computing Systems. With good, sound experience in C/UNIX, 68000 ASSEMBLER, 4GL/SQL, FORMAL METHODS or OPERATING SYSTEMS you can expect to break new ground in the Planning, Design, Implementation and Evaluation of Total Security Systems for the Government, Industry and Commerce sectors. Technically challenging, stimulating work in an advanced technology environment in the industries most exciting growth potential field. A generous salary, relocation, company pension, private medical insurance and BUPA complete the benefits package. MJ2565

*For more details on these opportunities call **MIKE JENKINS** on 0442 231691 office hours or 0582 456417 eves/wkends. Alternatively mail CV to Executive Recruitment Services. Hempstead House, Selden Hill, Hemel Hempstead, Herts, HP2 4LT or fax on 0442 230063*

ERS

HI TECH SOFTWARE to £27000

STS Recruitment specialises in placements within high technology companies. We currently have a large selection of SOFTWARE and Systems vacancies including the following:

IMAGE PROCESSING - London
Skilled 'C' programmers required to implement image processing networks. Applicants should have experience in developing network software products. Good growth prospects with dynamic company. Ref. 226/2413

MIS - W Country
Exciting information system development running on DEC hardware, requiring skills in 'C', SQL and graphics development (GKS). Pleasant working environment in West Country. 4-6 years experience required, including Structured Methodologies. Ref. 226/648

ISDN - Hants
At least 1 years experience in 'C' and UNIX within a datacomms environment needed for these software development positions based with a Blue Chip company. Rural location with excellent facilities. Ref. 226/2524

CONTROL SYSTEMS - Midlands.
Our client requires Software and Systems engineers to develop real time safety critical software for aerospace applications. 'C' and CORAL experience an advantage. Ref. 226/71

***STOP PRESS ** STOP PRESS ** STOP PRESS ** STOP PRESS**
PROGRAMMERS - Italy
An exciting opportunity for a number of experienced 'C' and UNIX programmers to work on the development of software systems for spacecraft control. Based in an attractive location close to ROME, our client can consider employment for a short term contract or on a permanent basis (including a relocation package to Italy). 2 years experience is required and a knowledge of C++, ADA and object orientated design would be advantageous. Ref.226/460

STS Recruitment has over 10 years experience in recruitment, so to secure your next career move contact us today send a detailed CV in complete confidence or telephone for further information quoting ref.XE/226

STS Recruitment

33 Staple Gardens,
Winchester,
Hants SO23 8SR

Tel: 0962 69478
Eve/Weekends: 0962 760241
Fax: 0962 841982

REAL-TIME CONSULTANCY

SURREY **to £25k + CAR**

My client, a small but well-established & expanding Systems Design Consultancy with excellent reputation in the International Aerospace Industry seeks Senior Software Engineer to head up new Software Dept.

Ideally in your mid-late 20's, with varied low & high level languages plus experience of Structured Techniques (eg ADA, JSP etc) & the ability to present solutions to Junior Engineers & Co. Directors alike. Tremendous career opportunity affording continual technical variety, UK & International travel - and the chance of playing a key role in shaping the Company's Software Development stategy.

Contact David Bird.

TRANSPUTER SUPPORT

Surrey **£24k + CAR**

Excellent opportunity for young professional Engineer to join dynamic, small commercial Company aiming to be the NO.1 UK Parallel Processing Systems Distributor.

Devise and configure your own systems, provide in-depth Technical Expertise to customers in the UK & overseas - and in a refreshingly exciting NON-MILITARY market-place!

Technical expertise at Device & Board level required, interest in Graphics & Super-Computing advantageous. Good inter-personal skills essential!

Contact Jim Fowkes.

Ring TODAY on 081-549 6441

59 Eden Street, Kingston, Surrey KT1 1BW.
Tel: 081-549 6441. Fax: 081-549 6462.

HIGH TECHNOLOGY RECRUITMENT ab Executive

LONDON & SOUTH

ORACLE/INGRES
London and Home Counties
to £27,000 + Car

Rapid increase in the use of ORACLE and INGRES throughout the computer industry means that your 6 months or more experience in one of the above 4GLs now carries a high premium. Opportunities exist in the following areas:
- Development
- Support
- Consultancy

You must be presentable, articulate and wish to work in a fast-moving environment. For immediate action send a full CV or telephone for further details.

SOFTWARE ENGINEERS
London - Thames Valley - Bristol
to £20,000

This leading systems house specialises in providing total IT solutions to the defence, industrial and communication systems sectors. In order to satisfy and ever increasing client base they now require talented Software Engineers with experience of at least one of the following:

Pascal — Assembler
ADA — Fortran
UNIX/C — 4GL's

These positions offer real scope for career advancement in addition to providing variety and challenge in a fast moving environment.

UNIX PRE-POST SALES
London, Home Counties
to £25,000 + Car

If you are currently working in a UNIX support role and are looking for the rewards you know you truly deserve then look no further. This leading manufacturer and systems house is looking for UNIX professionals with expertise in one of the following areas:
- OFFICE AUTOMATION
- COMMUNICATIONS
- 4GLs-RELATIONAL DATABASES

Excellent communication and presentation skills are essential. In return for your expertise you will be joining an elite team of UNIX professionals in a company that has been growing at 70% p.a. for the past 2 years.

INFORMIX/UNIFY
Support and Development
London, Home Counties
to £25,000 + Car + Bonus

This leading systems and software house is looking for committed professionals with at least 6 months experience in one of the above relational databases. A knowledge of UNIX is essential and any exposure to office automation/accounting packages would be useful. Opportunities exist in Pre/Post Sales and Development.
Telephone or write with a full CV.

MBA
Recruitment Consultants
Michael Bailey Associates Ltd.

For more information please contact:
Michael Garlick on *071-253 8413* (24 Hours)
or write with a full CV to:
Michael Bailey Associates Ltd,
Rococo House, 281 City Road,
London EC1V 1LA
quoting reference EXE/8/90
***Fax:* 071 253 6664**

ETHICAL RECRUITMENT & CONSULTANCY

We are specialists in applying ethical recruitment criteria to all areas of business, particularly in technology.

Our emphasis on SOCIALLY USEFUL TECHNOLOGY and our EQUAL OPPORTUNITIES approach produce a resource of exceptionally motivated and socially aware people, at all levels and in all disciplines.

Whether or not you wish to enhance your career by getting more of the same plus something extra, or are looking for a new angle on life - we will listen to you, and then to our client organisations and with your good will, make the match. More than occasionally inspiration occurs.

TRY US

EXCHANGE RESOURCES
Ethical work in technology

E-mail
(GreenNet)gn:exresource
GeoNet)geo2:exresource
Fax: 0225 469673

28 Milsom Street
Bath BA1 1DP
Telephone: 0225 469671

THE MICROSOFT PROGRAMMERS LIBRARY

Comprehensive CD-Rom database of 72 Microsoft technical manuals and best selling reference books that can be popped up in your favourite editor.
3000+ sample programs broken down into thousands of files.
Ideal for those learning as well as professionals.

THE BENEFITS

Reduce valuable programming time/meet deadlines.
Sample codes are indexed and referenced and can be copied or pasted directly into your editor.
Customised bookmarks allow you to define upto 80 'often used' reference points.
Development teams can be networked to one CD-ROM drive.
Use 'hot key' to 'pop up' for quick referencing.

THE REFERENCE CATEGORIES

- Windows • OS/2 • Macro Assembler(MASM)
- 'C' • MS-DOS • Pascal
- FORTRAN • BASIC • Hardware

THE SUPPORTED EDITORS

- BRIEF 2.0 • Epsilon 3.21 • QuickBasic 4.0
- Microsoft • Norton 1.3c • QuickC Editor 1.0
- Turbo Pascal • TurboC • VEDIT PLUS

LIBRARY CD £295 — CD DRIVE £595 — LIBRARY + DRIVE £799

All prices Exclude VAT

SOFTLoK
International Limited
Units 10 & 11, Railway Road Industrial Estate, Darwen, Lancashire, BB3 3EH, England.
Tel (0254) 772220 Fax (0254) 705956

CIRCLE NO. 178

THE UNIX RECRUITMENT SHOP

MANAGEMENT CONSULTANTS, Central London (£25k - £Neg)
- 8 + years Development & Management experience.
- Experience of OSI Architecture.
- Understanding of emerging Open Systems Technologies.
- Effective communications with people at all levels.
- Understand the latest concepts in Quality and the use of modern design tools.

SENIOR SOFTWARE ENGINEER, Portsmouth (£20k - £Neg)
- Commercial experience a must.
- Good interpersonal, presentation & project planning skills.
- Experience Informix 4gl essential.
- Experience of Databases | Office Automation | Networking.

EIS/MIS SUPPORT/CONSULTANT, Berks (£20k - £Neg)
- Excellent Communication Skills.
- Technical background in IBM, DEC or Unix Environment.
- High analytical and problem solving abilities.
- Willingness to travel overseas.

PROGRAMMERS, Middlesex (£11k - £16k)
- 3 + years programming experience in C.
- Working in an MS-DOS or Unix Environment.
- Applications programming of a commercial nature.
- Experience of EPOS.

Open Systems People Ltd is a technical recruitment agency specifically aimed at recruitment of staff at Technical Support, Systems Programmer, Project Leader and Middle Management levels. Our clients are involved in 'State-of-the-art' software engineering, in fields such as Electronic message handling, Document architectures, Relational Databases, LAN/WAN, Graphical User Interfaces, and other emerging technologies. If you have a strong background in Unix then make a move in the right direction and contact Open Systems People, *your technical friends in recruitment.*

Contact Hanif Noormohamed on
0767 317942 - Daytime or Evening
0767 315795 - Fax
for an informal discussion.
Alternatively fax or write.

OPEN SYSTEMS PEOPLE
PO Box 446 Biggleswade Beds SG18 0QZ

Unix is a registered trademark of AT & T

STOB - That divine spark

Verity answers your questions about artificial intelligence.

What is the goal of AI?

It is, of course, a vast simplification to think of the whole of the AI movement moving towards a single goal. There are in fact three. These are:

1) To produce a program that plays noughts and crosses infallibly.

2) To produce a program that plays chess infallibly.

3) To produce a working replica of Robbie the Robot, as seen in the 1950s sci-fi film 'Forbidden Planet'.

How far has it moved toward these goals?

The noughts and crosses one they have more or less cracked. Chess is proving a tougher nut. It was easy to reproduce the skill level of the average human player (gets the position of the rooks and the bishops the wrong way round, refers to the small doorknob-shaped pieces as 'prawns') or even the talented amateur (pretends to understand castling), but to achieve a true Grand Master level (genuinely interested in chess puzzles printed in Sunday Papers) is many years away yet.

Sad to say, the chances of duplicating Robbie the Robot are receding as time passes - the art of fashioning Perspex with such skill having been lost in the early 1960s.

What is the best example of a successful AI application?

The only successful example of an AI application was the ELIZA program (Weizenbaum, 1907). This was able to hold a dialogue with the operator in natural English.

Was that the silly program that pretended to be a psychologist?

Yes, it relied upon a system of identifying key phrases. The main disadvantage was that, whenever asked a difficult question, it tended to change the subject.

Wasn't ELIZA a bit of a cheat, then?

Why don't you ask me about AI languages?

Oh, all right. What about the AI languages?

There are three AI languages: LISP, Prolog and POP. They can be distinguished thus: LISP is the one with loads of brackets, Prolog(ue) is the European language with a US spelling and POP is the Great British Hope.

And what are the best-known AI applications of these languages?

The best-known AI application of LISP is CAD, such as AutoCad. The best-known application of Prolog is to buy a copy of Turbo Prolog in a rash moment, and leave it on the shelf collecting dust, always meaning to take it down and have a go. The best known application of POP is to be the subject of magazine articles entitled: 'POP - the British AI language.'

No, I meant what* programs *are written using them?

Only one program has ever been written using these languages: the 'Towers of Hanoi' game.

Surely the problem with AI languages is that they are too slow for commercial work?

There are certainly no flies on you. In 1988, a Respected Industry Commentator said that, in the next three years, hardware will speed up sufficiently to fix this.

So AI will be viable in 1991?

No, in 1993. AI will always become feasible in three years' time. This is an application of Carrollian logic programming - 'Jam tomorrow'. Remember, 95% of AI text books contain a quote from *Through the Looking Glass.*

Is there anything else I should know about AI?

An inference engine is a two-stroke version of a database engine. A knowledge engineer earns tons more money than you do. Backward chaining is still illegal in certain States of the US. Waltz's Algorithm is much less fun than its name.

Verity, thanks very much for your valuable time.

Always a pleasure for you, sweetie.

EXE

Opportunities for Software Professionals

REAL TIME

★ SYSTEMS PROGRAMMER
BERKSHIRE **to £17K**
Create new Run Time Systems using 8086/80286/80386 assembler and provide maintenance support to all OEM customers. Additional experience of three of the following is essential:- DOS, OS/2, COBOL, C, FLEX, UNIX or VM.
Ref/1630/05

★ SENIOR PROGRAMMER
SURREY **TO £16k**
Must be able to write small program specifications, have a logical approach to problem solving and have around three years experience in areas such as configuration control, CASE Tool support, coding and testing. ADA or PASCAL/CORAL 66 and VMS experience is necessary.
Ref/1504/01

★ SOFTWARE ENGINEERS
HAMPSHIRE **TO £24k**
To design & develop radio trunking systems and special customer projects involving the analysis of requirements, upfront task design, writing software, test & integration and system testing, educated to degree standard plus R/T experience using VAX/VMS.
Ref/1460/00

UNIX ✻ UNIX ✻ UNIX

★ OPEN SYSTEMS SPECIALISTS
LONDON, HERTFORDSHIRE, THAMES VALLEY **£20-£40K**
Consultancy, R+D and programming positions available. UNIX KERNEL and Shell experience with any of the following: C, C++ etc
Ref/00379/1079

★ C PROGRAMMERS
NATIONWIDE **TO £22K**
We continue to experience a massive shortfall in experienced C Programmers and can offer a variety of interesting and challenging positions. If you can develop, implement and test quality Software in C, under either UNIX or MS-DOS then we can probably find you the job you want.
Ref/1669/

CONSULTANCY / SUPPORT

★ POST SALES SUPPORT
BERKSHIRE **£15-20K**
Knowledge of manufacturing industry, in particular production control & inventory management. Software experience of RPGIII, IBM AS400, IBM 36/38. Duties will include bug fixing, application development and implementation. Travel possible.
Ref/1381/05

★ CONSULTANT
SURREY **£15-20K**
At least 10 years experience with a thorough computer science background. Should possess experience of small project management with real time skills using ADA, PASCAL, CORAL 66 or VMS.
Ref/1501/01

★ PRE/POST SALES SUPPORT
LONDON & COUNTRYWIDE **£16-27K + Car**
UNIX System manufacturers require high calibre professionals to move into Applications/Consultancy. Excellent technical and commercial skills a must for front line role in blue chip companies.
Ref/00657

★ NETWORK APPLICATIONS SUPPORT
BERKSHIRE **TO £23K + Car**
This position involves installation of PC Networking applications and its support, network security and database maintenance. Experience of NOVELL NETWARE, LOTUS, EXCEL and WordPerfect is required along with some exposure to PROGRESS 4GL.
Ref/1671/05

MANAGERIAL

★ SOFTWARE ENGINEERING TOOLS MANAGER
HAMPSHIRE **C£23k + Car**
Minimum 10 years experience (preferably software house background). Extensive knowledge & use of software engineering tools - automate, exelerator, software through pictures, information engineering workbench, Yourdon workbench.
Ref/1433/00

MIS/DATABASE SYSTEMS

★ TECHNICAL CONSULTANTS
LONDON **TO £14-20K**
Experience of applications design and development using Informix or ORACLE. Cross training can be provided with experience of other open systems rdbms. This will be an analyst/programmer role involving systems design and programming, client contact and the use of SSADM, Jackson and Yourdon.
Ref/1005/12

★ SYSTEMS ANALYST
BERKSHIRE **TO £20K + European Travel**
At least seven years analysis and COBOL programming experience required to help design and build a new European integrated package for this International company. Honeywell experience is advantageous as is the ability to speak any European language.
Ref/1670/05

★ ANALYST PROGRAMMER
NORTHAMPTONSHIRE **£10-14K**
To support existing applications for supplied & bespoke software, analyse business requirements, including the development of bespoke systems, installation of new hardware & software. Knowledge of LANs, spreadsheets, dBase IV, COBOL useful.
Ref/1494/55

ARTIFICIAL INTELLIGENCE

★ KNOWLEDGE ENGINEERS
HOME COUNTIES **£15-25K**
Rapidly expanding Expert Systems Software house requires young developers to design and implement projects. 1 years experience minimum of any AI language or C, and UNIX etc.
Ref/01699

GRAPHICS/CAD/DSP

★ CAD DEVELOPMENT ENGINEERS
BERKSHIRE **£15-19K**
High level language programmers from a CAD/CAE or similar background needed to develop suites of CAD tools for ASIC/VLSI state of the art environment. UNIX, C, PASCAL, LISP. Knowledge of industry standard tools an advantage.

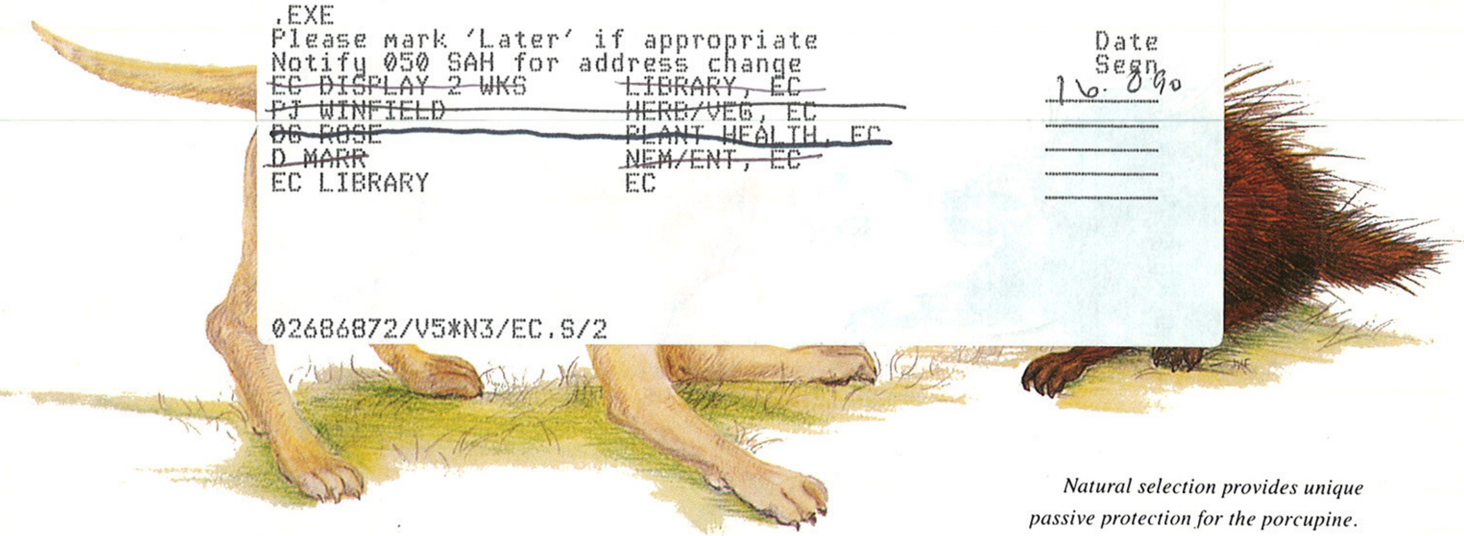
For further information on these or related positions call
ANITA HARVEY ON (0734) 774234 or (0734) 790928 after 7p.m.
FAX: (0734) 772773
Or write in confidence to CPS at:-

63 Peach Street
Wokingham
Berkshire RG11 1XP

CPS
COMPUTEC PERSONNEL SERVICES

Available for MS-DOS and Macintosh

.EXE
Please mark 'Later' if appropriate
Notify 050 SAH for address change
~~EC DISPLAY 2 WKS~~ ~~LIBRARY, EC~~
~~PJ WINFIELD~~ ~~HERB/VEG, EC~~
~~DG ROSE~~ ~~PLANT HEALTH, EC~~
~~D MARR~~ ~~NEM/ENT, EC~~
EC LIBRARY EC
Date
Segn
16.8.90
02686872/V5*N3/EC.S/2



Natural selection provides unique passive protection for the porcupine.

The Activator - Natural Selection For Software Protection

Inventor and entrepreneur Dick Erett explains how "The Activator" provides sane protection for your intellectual property.

"In any industry, just as in nature, the process of natural selection raises one solution above another. Natural selection is the most elegant of engineers.

In the area of software protection The Block has been selected by the marketplace as the solution that works. Over 500,000 packages are protected by our device.

For the past 4 years our philosophy has been; *'You have the right and obligation to protect your intellectual property.'*

A New Ethic For Software Protection

In allowing end-users unlimited copies of a software package and uninhibited hard disk and LAN operation, The Block has created a new ethic for software protection.

By removing protection from the magnetic media we remove the constraints that have plagued legitimate users.

They simply attach our key to the parallel port and forget it. It is totally transparent, but the software will not run without it.

A New Technology For Software Protection

Our newest model, The Activator, builds on our current patented design, and establishes an unprecedented class of software protection.

We have migrated and enhanced the circuitry of The Block to an ASIC (Application-Specific Integrated Circuit) imbedded in The Activator.

This greatly improves speed and performance, while reducing overall size. Data protection can also be provided.

Programmable Option

The Activator allows the software **developer** the option to program serial numbers, versions, or other pertinent data known only to the developer, into the circuit, and access it from the program.

Once you program your part of the chip, even we have no way to access your information.

The ASIC makes emulation of the device virtually impossible. It also presents an astronomical number of access combinations.

Full 100% Disclosure

Since The Activator is protected by our patent we fully disclose how it works. Once you understand it, endless methods of protection become evident.

Just as no two snowflakes are the same, no two implementations of The Activator are identical. And like the snowflake the simplicity of The Activator is its greatest beauty.

We never cramp your programming style or ingenuity. Make it as simple or complicated as you desire.

Let us help safeguard what's rightfully yours. Please call today for additional information or a demo unit. *It's only natural to protect your software.*"

"On-the-fly" programmable memory option now available for OS/2, MS-DOS and Macintosh

Software Security UK LTD
21A The Precinct, High Street, Egham
Surrey TW20 9PN
TEL: (0784) 430060
FAX: (0784) 430050

Unlimited Copies • Programmable • No Batteries • Small Size • Fast • Patented • Data Protection

CIRCLE NO. 175